AMIGOWIEC

6/1992 Pismo użytkowników komputerów AMIGA cena 10.000 zł.

Amiga i muzyka
Hamulec do Amigi
Suplement do Virus Checkera
Obsługa wejścia i wyjścia



GURU

Commodore

EDITORIAL



I tak nastało lato...

Dla nas jest to okres tym bardzie gorący, ponieważ ostro zabraliśmy się do AMIGOWCA. Tu nowe coś tam, tam nowe tu coś. Oczywiście nie udaje nam się uniknąć różnych błędów (mam nadzieję, że drobnych). Ostatnio myślałem, że wyjdę z siebie, gdy zabrakło znaczka TAI-CHI. Zamieszczam go poniżej:



Teraz chyba jasne jest o co w tym wszystkim chodzi. Kto nie rozumie idei tego znaczka niech poczyta "Amigowe strachy" z poprzedniego numery.

Pragnę wszystkich uspokoić, że nie ma nowego pseudowirusa ADDAM> - jest po prostu <SADDAM>, a literki to my połknęliśmy i płoniemy z tego powodu ze wstydu.

Przejdźmy jednak do tego co dzisiaj w numerze.

Zamieszczamy dokładny opis GURU. Przyda się to wszystkim programistom, ale myślę, że chętnie przeczytają to również zwykli śmiertelnicy - zawsze warto wiedzieć, o co chodzi w tym napisie z czerwonymi ramkami.

Postanowiliśmy opisać przeróbkę sprzętową - hamulec do AMIGI. Być może wyda się to komuś dziwne, że wszyscy przyspieszają, a my tutaj hamujemy. Hamulec od czasu do czasu też jest jednak przydatny (patrz. TAI-CHI). Można wtedy np. powoli potrenować w jakimś etapie Lemingów, żeby go potem przejść "normalnie", można też spowolnić Amigę i zobaczyć jak działa nasz program. Inne zastosowania na pewno znajdziecie sami. Montaż na odpowiedzialność autora.

Ku przestrodze opisaliśmy "Historię jednego rozszerzenia". List ten otrzymaliśmy od rozgoryczonego czytelnika. Sam dodam w tym miejscu, że kiedyś pracowaliśmy na rozszerzeniach 1.8 MB VizRam i praktycznie była z nimi masa kłopotu - a to zegar przestawał działać (nootycznie) a to w końcu trzeba było gwarancyjnie wymienić kości. Wniosek z tego taki - jeśli coś kupujesz, to i tak kupujesz kota w worku.

Jeśli po zamontowaniu czegoś częściej niż zwykle wiesza Ci się Amiga to znaczy, że coś jest nie tak i nie ma co z tym fantem czekać.

Na koniec niespodzianka! Dysk książka dla wszystkich początkujących AMIGOWCÓW - wszystko to co znajduje się na dysku WorkBencha zostało opisane i teraz wystarczy kliknąć na odpowiednią ikonę by się dowiedzieć co ona zacz.

Czekamy na Wasze listy. W tej chwili powoli opanowujemy to co już do nas przyszło (efekty widać zresztą na łamach pisma) i sukcesywnie odsyłamy Wam to co Wasze. Za wszelkie opóźnienia serdecznie przepraszamy.

Życzę przyjemnej lektury
Tomasz Kokoszczyński

PS. Nie zapomnijcie podpisać swoich dyskietek pełnym adresem!

SPIS TREŚCI

AMIGOWIEC

Czerwiec 1992 cena 10 000 zł

WYDAWCA:
ALFIN
TOMASZ KOKOSZCZYŃSKI

ADRES REDAKCJI:
P-17
85-099 BYDGOSZCZ 23

KONTO: PKO I BYDGOSZCZ, 9511-64839-136

Redaktor naczelny: Tomasz Kokoszczyński

Z-ca redaktora naczelnego: Ryszard Kowalski

Kolegium redakcyjne: Tomasz Kulbacki, Krzysztof Nowicki, Krzysztof Wirszyłło, Maciej Klimkiewicz, Łukasz Bieńkowski, Dariusz Zwierzyński, Jacek Ryć, Tomasz Flanc, Wojciech Białkowski,

Korektor: Aldona Ossowska

Redakcja nie odpowiada za treść reklam i ogłoszeń

1 moduł reklamowy kosztuje 90 000 zł

Ogłoszenia drobne tylko na blankiecie AMI-MARKETu

Commodore i *AMIGA* są znakami towarowymi firmy Commodore Electronics Ltd.

Skład i łamanie: Bydgoska Agencja Wydawniczo-Reklamowa „PAIWAN"
Druk: HECTOR

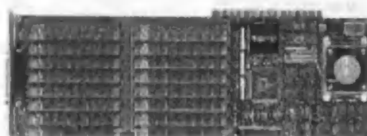


DEPESZE CZERWIEC'92

Workbench 2.1
Firma Commodore przygotowuje się do wypuszczenia na rynek nowego Workbench'a oznaczonego 2.1. W tej chwili krążą po kraju wersje beta tego systemu. Aby móc z niego skorzystać komputer musi posiadać Kickstart 2.04. Najważniejszą zmianą w stosunku do wersji poprzedniej jest wyposażenie go w możliwość wyboru języka w jakim system ma się z nami komunikować (nie ma niestety wśród dostępnych języków polskiego). W nowym Workbench'u zrezygnowano z MountList'y a zamiast niej wprowadzono do systemu zastępujący ją katalog o nazwie "Storage". Dodatkowo otrzymujemy na dyskach instalacyjnych program CrossDOS służący do przenoszenia plików pomiędzy dyskami zapisanymi w formacie Amigi i IBM lub Atari ST.

68040 do ataku !
Na rynku kart turbo do wszelkich Amig panuje straszliwe tempo. Firmy produkujące karty z procesorami 68020 i 68030 obniżają dość szybko ceny, na horyzoncie pojawiają się bowiem jeszcze szybsze karty z procesorami 68040. W tej chwili są już dostępne karty Zeus firmy Progresive Peripherals & Software wyposażone w te procesory taktowane zegarem 28 MHz (w zapowiedziach 33 MHz). Jest to karta przeznaczona dla Amigi serii 2000 zintegrowana z kontrolerem SCSI-2 oraz rozszerzeniem pamięci 32 bitowej (max 64 MB). Dla Amigi 500 i 500 + przeznaczona jest karta Progresive 040/500 wyposażona (oprócz procesora oczywiście) w rozszerzenie pamięci do 4 MB, Kickstart 2.xx oraz układ FASTROM pozwalający na skopiowanie Kickstart'u (w przypadku korzystania z systemu 1.3) do 32 bitowej pamięci. Istnieje także wersja tej karty z procesorm zintegrowanym z kooprocesorem matematycznym. W ofercie firmy znjduje się także propozycja dla posiadaczy Amigi 3000. Karta przeznaczona dla posiadaczy tego komputera wyposażona jest w procesor 68040 zintegrowany z kooprocesorem matematycznym oraz nowy układ zarządzający pamięcią, która może być rozszerzona na karcie do 32 MB. Cena takiej przyjemności - około 1400 funtów.





Geografia może być ciekawa.
O czym możemy się przekonać posiadając Amigę CDTV i program o nazwie American Vista Atlas. Dysk zawiera dokładne mapy USA, wiele ciekawych informacji statystcznych dotyczących demografii i ekonomii oraz historii tego kraju. Na dysku znajduje się także ponad 1000 zdjęć archiwalnych oraz wiele przykładów muzyki ludowej.

Pomuzykujmy
Firma Sunrize wypuściła na rynek 12 bitową kartę dźwiękową wyposażoną w sampler. Dostarcza się ją wraz z specjalnym programem o nazwie „Studio 16". Karta i jej oprogramowanie pozwala na dygitalizację dzwięku z szybkością próbkowania do 100 kHz. Wczytywany dzwięk może być zapisywany do pamięci lub na dysk twardy. Karta wyposażona została w specjalne filtry pozwalające na eliminacje niektórych z błędów dygitalizacji. Dołączone oprogramowanie pozwala na synchronizacje otwarzanego dźwięku z obrazem video. Oczywiście możliwe są także wszelkie operacje na samplach dostępne w standardowych programach, które służą do ich edycji. Firma Sunrize planuje wypuszczenie na rynek karty 16 bitowej (oznacza to dorównanie płycie kompaktowej).

Niech żyje król.
Jeżeli denerwuje cię zbyt wolna praca Arexx'a musisz koniecznie zaopatrzyć się w kompilator tego języka opracowany przez Dineen Edwards Group o nazwie REXX PLUS Compiler. Skompilowane aplikacje Arexx'a pracują od dwóch do piętnastu razy szybciej. Są one rezydentne co jeszcze bardziej skraca czas oczekiwania na efekty pracy makroinstrukcji. Kompilator jest dostarczany na dwóch dyskietkach z 270 - stronicową instrukcją oraz dodatkową biblioteką o nazwie rexxplsextnd.library zawierającą nowe instrukcje tego ciekawego języka.

Nauka fizyki na Amidze.
Physi-Calc jest nowym programem do nauki fizyki. Składa się z dwunastu modułów symulacyjnych mających służyć nauczycielom do demonstracji takich zjawisk fizycznych jak grawitacja, oscylator harmoniczny, ruch po okręgu. Wraz z programem dostarczany jest przetwornik analogowo - cyfrowy służący do podłączenia wszelkiego typu czujników labaroatoryjnych. Pozwala on wraz z oprogramowaniem na rejestrację i analizę eksperymentów fizycznych.

Program dla zapominalskich.
Dla wszystkich, którzy nagminnie gubią notesy z zapisanymi numerami telefonów, datami spotkań oraz notatkami firma Consultron przygotowała pro-



gram The Contact. W porównaniu do innych programów tego typu charakteryzuje się on dwiema zaletami. Po pierwsze: jednemu nazwisku można przypisać więcej niż jeden telefon. Po drugie program współpracuje z Arexx'em co umożliwia wykorzystanie jego możliwości w innych programach np. do drukowania jednego listu do kilku adresatów, których dane znajdują się w bazie danych. Jednym z modułów tej aplikacji jest programowalny, naukowy kalkulator, który bije na głowe kalkulator znajdujący się na dyskietce systemowej.

A może jakiś krajobrazik.

W dalszym ciągu trwa walka pomiędzy dwoma programami do tworzenia krajobrazów czyli Vistą i Scenery Animatorem. Natychmiast po ukazaniu się nowej wersji Visty na rynek trafiła druga edycja Scenery Animatora. Pierwszy z tych programów został wyposażony w możliwość wkomponowania w obraz dróg oraz wszelkiego typu roślinności. Dodatkowo można w nim uzyskać efekty wschodu i zachodu słońca. Autorzy drugiego z nich rozszerzyli możliwości swojego programu tylko o definiowanie roślinności. Wydaje się jednak, że drugi z tych programów wygra rywalizacją szczególnie w zastosowaniach domowych, albowiem przy znacznie niższym nakładzie pracy można uzyskać ciekawe efekty. Ponadto wyniki jego pracy, w przypadku posiadania Amigi bez wzmocnionych możliwości graficznych, są nieco lepsze (moim zdaniem).

Nowinki w C.

Jeżeli nie możesz sobie poradzić z programowaniem grafiki, menu lub gadget'ami musisz koniecznie wyposażyć się w program IntuiGen 1.0. Generuje on następujące struktury NewScreen, NewWindow, Menu, MenuItem, IntuiText oraz Gadget.

Co nowego w Imagine.

Ten interesujący program 3D staje się coraz bardziej popularny. Świadczy o tym coraz większa ilość obiektów, tekstur, efektów specjalnych, które przeznaczone są dla tego programu. Wychodząc na przeciw wszytkim tym, których denerwuje metoda uzyskiwania krzywych w tym programie firma Technical Tools opracowała programik o nazwie Lissa, który pozwala w prostszy sposób tworzyć obiekty tego typu.

Imagemaster

Jest to program przeznaczony dla użytkowników potrzebujących programu do obróbki 24 bitowych obrazów. Szególnie przydatny będzie tym wszystkim, którzy zajmują się obróbką zdjęć. Największym atutem tego programu jest możliwość dokonywania metamorfoz i zniekształceń w grafice. Podczas dokonywania tych zmian cały czas możemy kontrolować zarówno geometrię jak i przemianę kolorów podczas dokonywanej metamorfozy. Przykładem zastosowania programu tego typu mogą być ostatnie teledyski Michaela Jacksona. Oprócz wyżej opisanych możliwości program oferuje także standardowe funkcje graficzne malowania, komponowania oraz analizy obrazu.

Emulatory

Dla wszystkich posiadaczy karty A2286 AT Bridgeboard (dla niewtajemniczonych - jest karta emulatora IBM AT) mamy ciekawą informację. Firma ATOP wypuściła do tej karty rozszerzenie z procesorem 80386 (20 MHz). Rozszerzenie wykonane jest w formie płytki wkładanej w gniazdo procesora 80286. W kategorii emulatorów programowych także nowość. Pojawił się emulator CrossPC sygnowany przez firmę Consultron. Wymaga 1 MB pamięci MS-DOS'a w wersji 3.0 lub wyższej. Jest wstanie emulować kartę CGA. Rozpoznaje stację dysków Amigi jako 720 kB stacje IBM. Wszystki porty zewnętrzne Amigi widziane są jako porty IBM. Możliwa jest także współpraca z dyskiem twardym (partycja sformatowana w standardzie MS-DOS).

Amiga wspomaga lekarzy.

Wiele dzieci na świecie cierpi na chorobę zwaną dysleksją. Jej objawami są trudności w prawidłowym odczytywaniu tekstów. Dzieci mające dysleksję mają w związku z tym problemy z uczeniem się oraz nastręczają wiele problemów wychowawczych. Leczenie tej choroby wymaga wielu ćwiczeń polegających na wpajaniu uczniowi prawidłowych ruchów gałki ocznej. W brytyjskim oddziale firmy Commodore jest dostępny program wspomagający leczenie tej choroby. W tej chwili wspomaga on naukę czytania. Jeżeli spotka się z zainteresowaniem jego funkcje zostaną rozbudowane o naukę gramatyki oraz korzystania ze słowników. Dla szkół posiadających Amigi program ten jest dostarczany niodpłatnie.

HotHelp - pomoc dla Ciebie

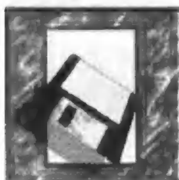
Firma MAXON znana z tak wielu wspaniałych programów jak np. MAXON-CAD, CHAMELEON i innych pokusiła się ostatnio o przeniesienie na Amigę przyszłościowego systemu pomocy. HotHelp jest jednocześnie podręcznikiem i leksykonem, a przy tym czujnym partnerem.



System ten działa na podobnej zasadzie jak Help w Windowsach. W każdej chwili dostępny, czy to z CLI czy też z Workbencha, pozwala na wyszukanie niejasnego hasła. Przy tym możemy go również zapytać o jakieś niejasne słowo w wyjaśnieniu itd. aż w końcu dowiemy się wszystkiego co chcielibyśmy wiedzieć. Takie wyszukiwanie jest o wiele szybsze niż „przewalanie" wielotomowych ksiąg i opracowań. Jest to system przyszłości, dzięki któremu, jeśli będzie obejmował odpowiednio duży zasób wiedzy, będziemy mieli dostęp do całej wiedzy jaka została zgromadzona przez ludzi. Jak na razie HotHelp obejmuje jedynie informacje dotycznące Amigi. Każdy może tu znaleźć coś dla siebie. Dla początkujących przewidziano dział SŁOWNIK i SHELL, zaś dla programistów mamy LIBRARIES & DEVICES + ANSI-C.

Cena takiego podręcznika wynosi 89 DM.

Pragnę w tym miejscu nadmienić, że choć nasz dodatkowy dysk PD oznaczony jako "B" jest tylko namiastką takiego systemu, z pewnością znajdzie uznanie wśród początkujących amigowców.



AMIGA I MUZYKA

BatBeast

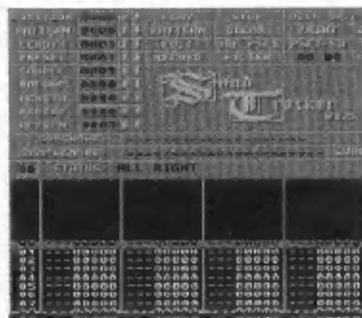
Komputer Commodore Amiga słynie ze swych niebywałych możliwości w dziedzinie kształtowania i przetwarzania dźwięku. Jednak przeciętny użytkownik, bez pomocy odpowiedniego oprogramowania, nie jest w stanie wydobyć ze swojego komputera żadnego, cichutkiego nawet szmeru.

Paleta programów muzycznych napisanych dla komputera Commodore Amiga jest tak ogromna, że pozwala praktycznie każdemu, związanemu z muzyką profesjonalnie lub nie, na tworzenie utworów muzycznych.

Niniejsza publikacja adresowana jest do tych użytkowników Amigi, którzy pragną wykorzystać wspaniałe możliwości dźwiękowe tego komputera. Należy jednak zaznaczyć, że nie stanowi ona elementarza dla początkujących muzyków, ale jest próbą krótkiego przedstawienia i porównania popularnych programów muzycznych dostępnych na Amigę. W zasadzie artykuł ten dotyczy tylko jednej grupy programów, a mianowicie "Trackerów".

Stanowią one obecnie największą grupę programów muzycznych. Zaliczyć do nich można wszystkie wersje programów Sound-, Noise- , Pro- tracker oraz najnowszą mutację: StarTrekker. Wszystkie te programy wywodzą się od wspólnego przodka, którym jest SoundTracker. Nowatorski program oryginalnie napisany przez Karstena Obarskiego doczekał się wielu wersji i przeróbek. Przeznaczony jest przede wszystkim dla ludzi zajmujących się pisaniem muzyki do gier i programów demonstracyjnych. Jedną z wielu zalet tego programu jest fakt, że do jego obsługi nie potrzebna jest znajomość zapisu nutowego - wystarczy jedynie wiedzieć, który klawisz gra jaką nutę, posiadać odrobinę wyobraźni muzycznej oraz zdolności manualnych i już można swobodnie poświęcić się komponowaniu. Poza tym program jest pełnowartościowym sequencerem o wielu możliwościach i dużej swobodzie obsługi. Pozwala na używanie do 15 (w starszej wersji) lub 31 sampli i instrumentów. Programem, który wyparł SoundTrackera (głównie dzięki swoim możliwościom) jest NoiseTracker. Napisany przez dwóch zdolnych programistów ze Szwecji o pseudonimach "Mahonej" i "Kaktus" znanych również ze wspaniałych dysków muzycznych. Właściwie program ten wygląda bardzo podobnie do poprzednika, a jego pierwsze wersje są, szczerze mówiąc, prymitywne. Dopiero jedna z ostatnich mutacji (wersja 2.0) daje nam do dyspozycji szereg bardzo pożytecznych narzędzi, takich jak: wbudowany moduł obsługi samplera czy interfacu MIDI.

Zanim przystąpi się do pracy z poważnym programem użytkowym, należy zapoznać się nie tylko z jego możliwościami, ale także z wymogami jakie stawia on przed użytkownikiem. Teraz postaram się przybliżyć Tobie, drogi czytelniku, kilka wersji "Trackerów".



SOUNDTRACKER 2.3

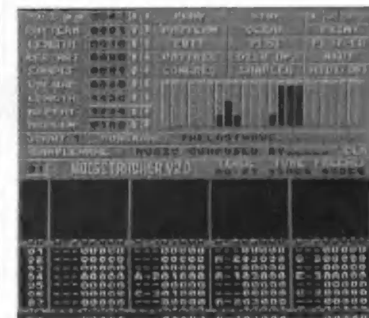
Ogólnie rzecz biorąc program SOUNDTRACKER 2.3 jest jeszcze bardzo niedoskonałym (żeby nie powiedzieć prymitywnym) narzędziem. Razi w nim przede wszystkim brak opcji LOAD SAMPLE i PRESET EDITOR co w znacznym stopniu utrudnia (a niekiedy nawet uniemożliwia) wczytywanie nowych instrumentów (a ich długość ograniczona jest do wartości 7FFE Kb).

SOUNDTRACKER 2.4

Posiada dokładnie to samo co poprzednia wersja (2.3) oraz dodatkowo kilka ciekawych i bardzo pomocnych rozwiązań. Po pierwsze maksymalna długość instrumentu wzrosła do wartości FFFC (64 Kb). A po drugie edytor wzbogacony został o TIMER - czyli zegar czasu rzeczywistego, FILTER - programowy filtr dźwięku i PSET-ED (!) - czyli to, czego w poprzedniej wersji brakowało nam najbardziej. PRESET EDITOR - służy do wpisania na dyskietkę pliku zawierającego listę nazw instrumentów, które posiadamy w swoich zbiorach. W pierwszych wersjach SoundTrackerów dodawany był jedynie jako oddzielny program pomocniczy. W SoundTrakerze 2.4 został wbudowany jako moduł stanowiąc integralną część programu.

SOUNDTRACKER 2.5

Kolejna "mutacja" SoundTrackera staje się coraz bardziej przyjazna człowiekowi.



NOISETRACKER 1.0

Programy z rodziny NoiseTrackerów są w pełni zgodne z SoundTrackerami. Oznacza to, że z poziomu NoiseTrackera można bez żadnych problemów



wczytywać, uruchamiać i modyfikować utwory napisane wcześniej pod SoundTrackerem. Należy jednak dodać (UWAGA !!!), że "twierdzenie odwrotne nie jest prawdziwe", a to ze względu na pewne różnice w zapisie utworów napisanych przy pomocy NoiseTrackera.

Noisetracker 1.0 (pierwszy z nowej rodziny) posiada kilka nowych, bardzo przydatnych rozwiązań. Nie wszystkie jednak pomocne narzędzia znane z SoundTrackera znalazły w nim zastosowanie (np. zegar). Ogromnie pomocny w pracy jest licznik pokazujący długość utworu (MODUŁU) w kilobajtach oraz ilość wolnej pamięci. Poza tym za pomocą klawisza [Help] możemy wejść do PLST EDITORA, a [Caps Lock] umożliwia nam zmianę trybu pracy w zależności od wykorzystywanego monitora (mono/color). NoiseTracker 1.0 posiada jeszcze opcję USE PSET i bardzo prymitywny (uniemożliwiający praktycznie pracę) PRESET EDITOR, ale i tak większość użytkowników wczytuje instrumenty za pomocą LOAD SAMPLE.

NOISETRACKER 1.1

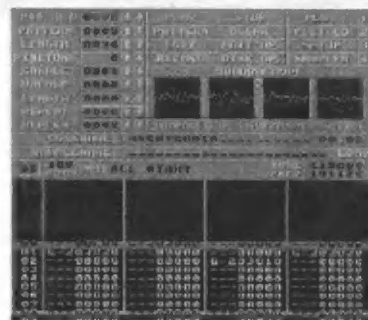
W tej wersji programu odstąpiono całkowicie od wczytywania instrumentów przez USE PSET pozostawiając jedynie opcje LOAD SAMPLE. W oknach określających ułożenie patternów pojawia się nowa opcja RESTART, dzięki której możemy zapętlić nasz utwór od dowolnie wybranej pozycji. Niewiele zmian, ale przecież "nie od razu Kraków zbudowano..."

NOISETRACKER 1.2

Już na pierwszy rzut oka można zauważyć szalone zmiany, jakie zaszły w tym programie. Zastosowano szereg nowatorskich rozwiązań, jak np. SCOPE (zamiast starego SPECTRUM ANALYZERa), który ilustruje nam przebiegi dźwięku podobnie jak oscyloskop. Pojawił się wreszcie zegar i szereg nowych opcji. Najpiękniejszą jednak sprawą w NoiseTrackerze 1.2 jest EDYTOR ŚCIEŻKI (!!!). Poprzez naciśnięcie [Ctrl] i odpowiedniego klawisza (opisane w opcji HELP) możemy dokonywać modyfikacji fragmentów ścieżki ! Jest to bardzo przydatne w praktyce, np. w sytuacji kiedy trzeba przesunąć ciąg 20 nut o 2 jednostki w dół lub w górę. Wystarczy wówczas zaznaczyć interesujący nas fragment ścieżki i stosując odpowiednią kombinację klawiszy dokonać przesunięcia.

NOISETRACKER 2.0

To jest to! NoiseTracker 2.0! Pojawienie się tego programu można uznać za przełom w historii "trackerów". Dzięki wbudowanej obsłudze SAMPLERa i interfaceu MIDI praca z nim nabiera nowego wymiaru. Poprzez sampler można wczytać do pamięci komputera dowolny dźwięk, a midi pozwala rozszerzyć możliwości muzyczne Amigi o nowe brzmienia i w pełni wykorzystać posiadany przez nas instrument. Obecnie w prawie każdym demie (programie demonstracyjnym) zawarta jest muzyka zrobiona właśnie przy pomocy tego programu. W osobnej przeróbce NoiseTrackera 2.0 zwanej SterTrekkerem (lub NoiseTrackerem 2.1) programista umieścił możliwość komponowania muzyki na ośmiu kanałach ! Oczywiście w programie zawarta jest pełna obsługa interfaceu MIDI i Samplera.



PROTRACKER 1.0

To program bardzo podobny do NoiseTrackera 2.0. Posiada jednak znacznie rozbudowane opcje MIDI, bardzo dobry PRESET EDITOR, a nowe rozkazy umożliwiają także formatowanie dyskietki i zmianę nazwy plików (RENAME). Potrójne menu EDIT OP. daje nam dużo nowych możliwości.

PROTRACKER 1.1b

Szokuje zaraz po uruchomieniu. Posiada SPECTRUM ANALYZER i QUADROSCOPE, kytóry wyświetla każdy kanał oddzielnie. Edytor wzbogacony o możliwość ręcznej zmiany tempa. Nowe możliwości DISK OP. pozwalają na odczytywanie dźwięków zapisanych w standardzie RAW i IFF. Posiada własny, całkiem odmienny od poprzednich, wielki na pół ekranu edytor SAMPLERa. Umożliwia on edycję całego SAMPLEa lub tylko jego wycinka (podobny do Audio Mastera).

STARTREKKER 1.0

Czyli NoiseTracker 2.0 wzbogacony o możliwość pracy na 8 kanałach. Podczas pracy w trybie "8" każdy pattern teoretycznie posiadać powinien 8 ścieżek. Tak jednak nie jest, a na ekranie edytora widać tylko 4 kanały. Przełączenie patternu na dalsze 4 kanały następuje po naciśnięciu klawisza tabulatora. Reszta opcji właściwie pozostaje bez zmian, dlatego jeśli pragniesz bliżej poznać ten program, przeczytaj najpierw fragment dotyczący NoiseTrackera 2.0.

STARTREKKER 1.2

Podczas pracy w trybie osimiokanałowym na ekranie widać 8 ścieżek. Jednak fatalna jakość dźwięku nie zachęca do komponowania w tym trybie. Bardzo ciekawie rozwiązano opcje DISK OP. - przejście do CLI i komunikacja przy pomocy specjalnego okna. Zerwano dzięki temu z rygorem przechowywania modułów i innych danych w katalogach o ściśle określonych nazwach.

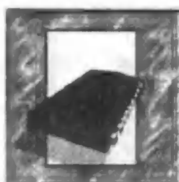
Ciekawostką może być również edytor dźwięku FM i AM, dzięki któremu, nie używając samplera, a jedynie zmieniając wartości odpowiednich parametrów, uzyskać można całkiem ciekawe efekty.

StarTrekkery to początek nowej ery! Są to "programy-kombajny" wyposażone we wszystko (no prawie) co potrzebne jest muzykowi.

Czy wiesz do czego służy SAMPLER ?

Amiga Audio Sampler jest aktualnie najpopularniejszym w naszym kraju urządzeniem przetwarzającym sygnał analogowy z wejścia na sygnał cyfrowy dla komputera. Na jego popularność niewątpliwy wpływ wywiera stosunkowo niska cena (jest to prawdopodobnie najtańsza przystawka do Amigi). Obecnie na rynku znajdują się dwie wersje tego urządzenia: V1.MONO i V2.STEREO. Instalacja samplera jest bardzo prosta, ale pamiętać trzeba o kilku podstawowych sprawach. Urządzenie należy podłączyć do portu równoległego typu CENTRONIX, który znajduje się obok gniazda zasilania. Operacji tej można dokonywać tylko przy WYŁĄCZONYM KOMPUTERZE. Inne postępowanie może zakończyć się uszkodzeniem komputera lub samplera. Gniazdo, które znajduje się w górnej części samplera przeznaczone jest do doprowadzenia sygnału analogowego np. z magnetofonu. W urządzeniu analogowym sampler dołączamy do gniazda słuchawkowego lub jeśli urządzenie nie jest w nie wyposażone do gniazda głośnikowego. Drugie rozwiązanie nie jest bezpieczne i zaleca się postępować tak tylko w ostateczności. Zbyt głośne nastawienie urządzenia może spowodować uszkodzenie samplera (ewentualność ta nie istnieje w pierwszym przypadku).

Sampler współpracuje z takimi programami jak: AUDIOMASTER, NOISETRACKER, itd. Umożliwia on tworzenie własnych efektów dźwiękowych, dem muzycznych, "przegrywanie" muzyki z magnetofonu na komputer. W celu prawidłowej dygitalizacji po uruchomieniu programu należy wybrać opcję monitora dźwięku i ustawić najlepszą dynamikę dygitalizacji regulatorem siły głosu w urządzeniu odtwarzającym, tak aby sygnał był jak najsilniejszy lecz nieprzesterowany (bez zniekształceń).



HAMULEC

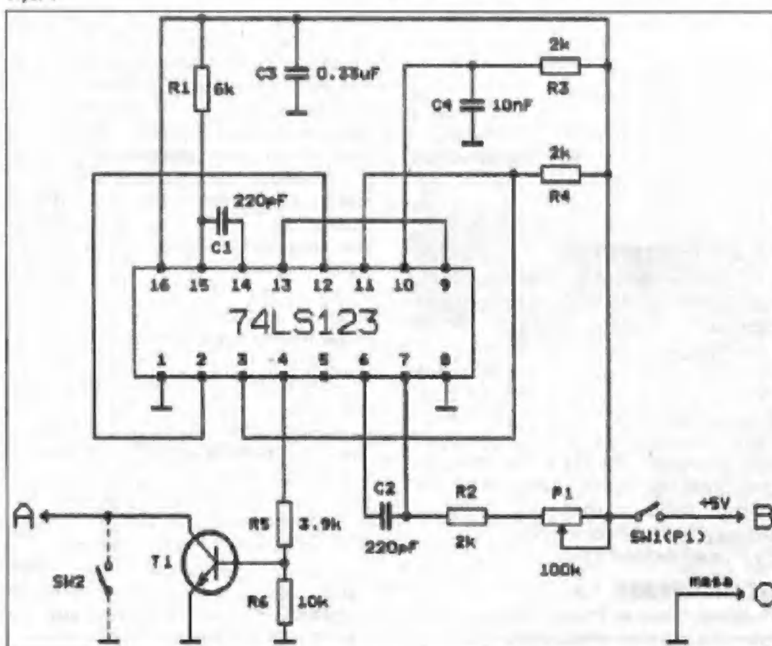
Jan Komarnicki

Amiga 500 nie należy do najszybszych komputerów, ale czasami odczuwamy potrzebę zwolnienia tempa jej pracy.

Poniżej zostanie opisany prosty układ elektroniczny, montowany wewnątrz obudowy Amigi, który umożliwia przyhamowanie lub zatrzymanie naszego komputera w dogodnej dla nas chwili. "Hamulec" może być wykorzystany przez miłośników gier do przebrnięcia szczególnie trudnych miejsc, wymagających ostrego wahlowania dżojstikiem, oraz przez hakerów, którzy nie posiadają jeszcze przystawki "Action Replay".

Układ, którego schemat przedstawiono na rys.1, umożliwia regulację szybkości pracy procesora w zakresie od ok.85% do 5% szybkości standardowej (pomiar wykonano przy pomocy programu Sysinfo V1.98). Zasada pracy układu polega na cyklicznym zatrzymywaniu procesora na pewien okres czasu, do czego wykorzystujemy sygnał prostokątny o regulowanym wypełnieniu generowany przez multiwibrator 74LS123. Sygnał ten podajemy na wejście HALT (nóżka nr17) mikroprocesora 68000. Podanie 0 (stan niski) na to wejście powoduje przerwanie pracy procesora po zakończeniu aktualnego cyklu, natomiast podanie 1 (stan wysoki) uruchamia go ponownie. Ponieważ wejście HALT pełni również rolę wyjścia (w przypadku pojawienia się tzw. podwójnego błędu magistrali procesor sam podaje 0), a w dodatku jest ono wyprowadzone na złącze rozszerzające komputera (to z lewej strony), w układzie zastosowano tranzystor oddzielający T1. Jest to szybki tranzystor impulsowy. Do regulacji szybkości pracy procesora służy potencjometr P1 z wyłącznikiem SW1, poprzez który podajemy napięcie +5V zasilające układ. W czasie normalnej pracy Amigi nasz układ jest wyłączony i nie pobiera żadnego prądu. Po włączeniu układu za pomocą potencjometru P1 (a konkretnie sprzężonego z nim wyłącznika SW1) Amiga zmniejsza szybkość o 15%, dalsze zmniejszenie szybkości uzyskujemy kręcąc potencjometrem w prawo. W prawym skrajnym położeniu potenjometru uzyskamy 5% szybkości standartowej, innymi słowy nasz komputer będzie pracował 20 razy wolniej. Oczywiście w każdej chwili możemy wyłączyć nasz "hamulec" i powrócić do normalnej pracy. Przełącznik SW2 umożliwia całkowite zatrzymanie procesora. Pobór prądu przez układ wynosi ok.14 mA.

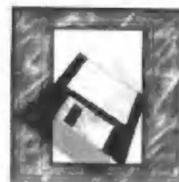
Rys. 1

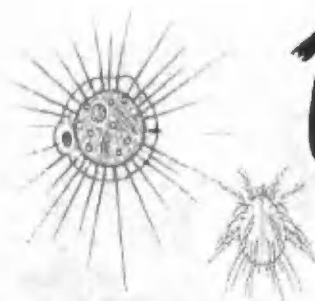


Układ można zmontować na zaprojektowanej przez siebie płytce lub też wykorzystać płytkę uniwersalną. Ja zastosowałem płytkę UM1, nabytą w składnicy harcerskiej i przyciętą do rozmiarów 27 x 40 mm.

Na rys.2 przedstawiono zasady montażu układu wewnątrz obudowy Amigi. Przewód A (wyjście naszego układu) łączymy do nogi nr17 mikroprpcesora 68000. Mało eleganckim, ale skutecznym sposobem jest przylutowanie przewodu bezpośrednio do nóżki układu scalonego. Przewód B (+5V) łączymy z pierwszą końcówką złącza rozszerzającego pamięć, przewód C (masa) z drugą, kolejną, końcówką tego złącza. Końcówki liczymy od dołu płytki, uwzględniamy tylko zewnętrzny (łatwo dostępny) rządek o dłuższych wyprowadzeniach. Przewody lutujemy tuż przy płycie głównej aby uniknąć zwarcia z metalową pokrywą ekranującą. Potencjometr mocujemy do kawałka kątownika i następnie przykręcamy do dolnej ścianki obudowy. Wcześniej musimy wywiercić otwór na ośkę potencjometru w ściance bocznej obudowy, co stanowi prawdziwą przyjemność ze względu na miękkość materiału. Płytkę z naszym układem mocujemy również do dolnej ścianki, jak na rysunku. Teraz mamy do rozwiązania najpoważniejszy problem: dobranie gałki, która pasowałaby kolorystycznie do obudowy Amigi, mnie niestety nie udało się tego rozwiązać.

Na zakończenie kilka uwag dla początkujących. Amigę rozbieramy wykręcając 6 wkrętów z obudowy, następnie zdejmujemy górną pokrywę, potem odłączamy klawiaturę (musimy zapamiętać jak była podłączona) i wysuwamy ją z zaczepów. Kolejna czynność to odkręcenie 4 wkrętów, odgięcie kilku metalowych zaczepów i zdjęcie metalo-





Virus Checker

suplement

Pieronek

Po przeczytaniu wzmianki w Amigowcu 3/92 o programie Virus_Checker doszedłem do wniosku, że chodzi o jego starszą wersję. Świadczy o tym fakt, że FixSaddam jest oddzielnym programem. Nowsza wersja ma FixSaddam wbudowany jako opcję i włącza ją automatycznie. Z Virus_Checkera, najpierw z wersji 5.26, teraz 5.33 korzystam już od pół roku. Pragnę podzielić się z czytelnikami swoimi spostrzeżeniami, zwłaszcza że oba programy mogą stwarzać pewne problemy. Najpierw powiem o starszej wersji. Z tej racji że FixSaddam jest oddzielnym programem możemy mieć problem z usunięciem niektórych wersji Saddama z dysku. W takim przypadku należy najpierw użyć FixSaddam. Wgrywamy Virus_Checker, umieszczamy FixSaddam w RAM: i wkładamy zarażony dysk. Po usunięciu intruza z pamięci będzie kilka komunikatów, przez które trzeba przejść i uruchomić FixSaddam. Po zakończeniu operacji naprawiania można już intruza usunąć, najlepiej przez wyjęcie i ponowne włożenie leczonej dyskietki oraz odpowiedzi na komunikat jaki się ukaże. Odradzam oddzielnego używania obu programów, może to być syzyfowa praca, ponieważ po naprawianiu, Saddam zawsze poprawi po swojemu. Virus_Checker bez FixSaddam nadaje się w zasadzie tylko do wykrywania Saddama. Przejdę teraz do nowszej wersji Virus_Checkera. Program posiada kilka nowych możliwości. Przy aktywnym oknie Virus_Checkera mamy możliwość sterowania z klawiatury:

- 0 - 3 sprawdzanie dysku, jak po włożeniu do stacji nr. 0 - 3
- m sprawdzanie pamięci
- s sprawdzanie na link/file
- f naprawianie po Saddamie

Podanie przy uruchamianiu Virus_Checker -n powoduje utworzenie okna 1x1 piksel położonego z tyłu za ostatnim otwartym oknem. W tym przypadku mamy niedostępne opcje aktywnego okna. Dostęp do nich jest możliwy z ARexx'a. Robi się to podobnie jak w starszej wersji. Do ARexx'a dodano polecenie 'scanforsaddam:DF0:; które uruchamia naprawiane efektów działalności Saddama. Opcja ta ma poprawiony kod fixsaddam, który powodował w pracy starszych wersji pewne przekłamania. Wyjaśnię to na przykładzie. Mam 3 egzemplarze Saddama. Virus_Checker 5.26 jeden z nich zaznacza tylko jako wymazany (DEL). Virus_Checker 5.33 tego nie robi, ale w dwóch przypadkach podaje komunikat ERROR Deleting File, co można poprawić przez wyjęcie, oraz ponowne włożenie leczonego dysku i odpowiedzi na komunikat. Już od wersji 5.30 ten kod jest poprawiony. Ta sprawa dotyczy systemu z procesorem 68000. Program nie posiada też wady, z powodu której po obejrzeniu bootblocku traciliśmy możliwość Learn. Dodatkowo program rozpoznaje bootblock Workbencha 2.0 i oczywiście jest 'mądrzejszy' o kilka wirusów. Liczba wirusów może być nieścisła. Opis nie jest pod tym względem zbyt wyczerpujący. Przykładowo Virus_Checker 1.0 nie rozpoznaje według opisu żadnych wirusów, czego raczej trudno jest oczekiwać. Dalsze zastrzeżenia Virus_Checkery mają wspólne. Zacznę od mniej przyjemnej sprawy. Obecność w pamięci bootselectora The Amiga Tool, który brany jest za Abrahama, powoduje wizytę Guru. To nie jest jedyny przypadek. Podobnie jest z System Z Protector, ale tylko w przypadku uruchamiania systemu z dysku z tym programem. Ze względu na istnienie wirusów o nazwie System Z Protector (HCS) i PVL Protector, możemy mieć wiele fałszywych alarmów. Radzę sprawdzić swoje zbiory pod tym względem. Dalsze sprawy już nie niosą za sobą takich konsekwencji, ale mogą powodować panikę. Możemy mieć problemy ze startup-sequence. Umieszczenie w niej nazwy programu ze spacją w środku lub nie zakończonej naciśnięciem RETURN, powoduje komunikat ERROR FINDING FILE. Przypomnę że sprawdzany jest tylko pierwszy plik, że startup-sequence (first file). To tyle zastrzeżeń. Programy te mimo wspomnianych zastrzeżeń są jednymi z najlepszych, oczywiście wersja 5.33 jest dodatkowo skuteczniejsza i wygodniejsza w użyciu. To dzięki tym programom, Saddam nie wyrządził mi żadnych szkód.

---

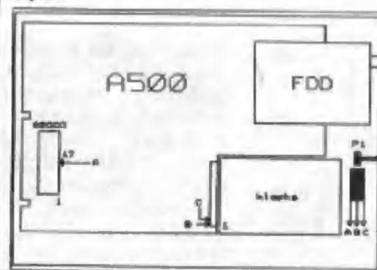
wej pokrywy ekranującej. Teraz możemy podziwiać wnętrze naszego komputera. Przy okazji warto zorientować się jaki rodzaj (revision) płyty głównej jest w naszym posiadaniu. Mówi o tym napis na płycie poniżej słów "B52 Rock Lobster". Amiga 500 miała dotychczas cztery podstawowe wersje płyt: rev3, 5, 6 i 7. Miłą niespodziankę może być fakt posiadanie Fat Agnus 8372A i możliwość rozbudowy pamięci chip RAM do 1MB na płycie głównej.

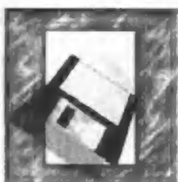
Przed podjęciem prac we wnętrzu komputera musimy pozbyć się ładunków elektrostatycznych: lutownicę, narzędzia i własną osobę należy rozładować przez uziemienie (można w tym celu dotykać niepomalowanych części kaloryfera). Należy używać lutownicy małej mocy, zasilanej z niskiego napięcia (mniejsze prawdopodobieństwo przebicia) i czas lutowania możliwie skracać, aby nie przegrzać elementów i nie uszkodzić ścieżek.

Wykaz części:
oporniki: R1=6k; R2,R3,R4=2k; R5=3.9k; R6=10k
kondensatory: C1,C2=220pF; C3=0.33uF; C4=10nF
tranzystor T1= BSX93, BSX94, BSX66, 2N2221, 2N2369, BSX59, 2N2218
układ scalony = 74LS123
potencjometr miniaturowy z wyłącznikiem P1= 100k B
piwo = 0.33l

Jan Komarnicki
01-684 Warszawa
ul.Klaudyny 32 m.178

Rys. 2





GURU I TY



Maciej Klimkiewicz

Za górami, za lasami, za siedmioma rzekami, przy stole z komputerem siedział pewien mędrzec. Wzrok jego utkwiony był w bursztynowej poświacie monitora. Radośnie migający napis „Guru Meditation" informował go o najgorszym. Wszystkie dane, nad którymi pracował od tygodni, zostały bezpowrotnie utracone. Tymczasem atmosfera w komnacie coraz bardziej zagęszczała się. Poziom humoru mędrca spadł poniżej zera. Latające przedmioty dobitnie świadczyły o powadze sytuacji...

Powróćmy jednak z wyimaginowanego świata bajki do rzeczywistości. Jak zapewne wielu z Was się domyśla, tematem tego artykułu jest „Guru Meditation". Myślę, że większość z Was (99%) wie o czym mowa. Nie każdy jednak orientuje się „z czym to należy jeść".

Zacznijmy od początku. System Amigi umożliwia wykonywanie wielu programów w tym samym czasie. Wygląda to w ten sposób, że procesor główny realizuje po kolei wszystkie programy po kawałeczku. Jeżeli w czasie sekundy procesor wykona kilka razy fragment każdego z nich, będziemy mieli wrażenie, że chodzą one równocześnie. Każdy taki chodzący programik nazywamy taskiem czyli zadaniem. Programiści też ludzie i podczas pisania programów popełniają błędy. Na komputerach, które nie umożliwiają wielozadaniowości, błąd kończy się zazwyczaj zawieszeniem systemu. Odrębnie sytuacja ma się na Amidze, na której może pracować kilka programów jednocześnie. Zastanówmy się, co się stanie, gdy któryś z nich zawiesi się?

I ZATRZYMANIE TASKU

Kiedy jeden z tasków Amigi z jakiegoś powodu blokuje się (np: komputer stracił kontakt z drukarką, a biedny procesor nie może przeżyć takiej straty) do akcji wkracza tzw. funkcja Exec. Jej zadanie polega na zatrzymaniu tasku, który jest sprawcą blokady. Dla nas widocznym tego efektem jest pojawienie się na ekranie Workbencha requestra z następującym napisem:

Software error - task held
Finish all disk activity
Select CANCEL to reset/debug

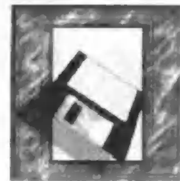
co można przetłumaczyć na polski w następujący sposób:

Błąd programowy - taks zatrzymany
Wszystkie dyski są aktywne
Wciśnij CANCEL
w celu zresetowania systemu

Komputer w dalszym ciągu działa, można więc nagrać na dyskietkę wszystkie potrzebne pliki, na których nam zależy, ale UWAGA !! Błędny task mógł narobić w systemie już tyle szkód, że po kilku sekundach Amiga zablokuje się ponownie, tym razem całkowicie (padnie system). Jeżeli w tym momencie będziemy nagrywali

tabela 1 — pole „a"

2	błąd sygnału wejściowego (błąd sprzętowy),
3	błędny adres (nieparzysta wartość komórki),
4	nielegalna instrukcja,
5	dzielenie przez zero,
6	błąd CHK (wartość rejestru poza zakresem),
7	rozkaz TRAPV (błąd przeniesienia),
8	użyty rozkaz uprzywilejowania względem programu zarządzającego,
9	moduł śledzenia,
A	emulator linii 1010 o kodach instrukcji od $A000 do $AFFF,
B	emulator linii 1111 o kodach instrukcji od $F000 do $FFFF.



coś na dysk, możemy stracić całą jego zawartość. Dlatego najlepszym rozwiązaniem jest nagrywanie danych na pusty nośnik. Gdy już wszystkie potrzebne dane mamy nagrane, możemy wcisnąć gadżet CANCEL. Będziemy mieli wtedy bliskie spotkanie III stopnia z Guru Meditation.

II ALARM

Jest to ostatnie „tchnienie" zablokowanego systemu. Najpierw przez kilka sekund miga dioda zasilania komputera, a następnie na ekranie pojawia się napis:
Software failute! Press left mouse button to continue. Guru Meditation #84010007.00C13870

Teraz nie pozostaje nam nic innego jak odszyfrowanie komunikatu w celu wykrycia błędu. Musimy przecież wiedzieć, co było powodem zawieszenia się programu, aby na przyszłość uniknąć tego samego błędu. Za napisem wyświetlony jest ciąg cyferek, który jest zaszyfrowanym kodem informacji. Generalnie błędy możemy podzielić ze względu na „winowajcę" na dwie grupy:

1) błędy procesora,

Błędy spowodowane przez procesor ukazują się w następujący sposób: Guru Meditation #0000000a.bbbbbbbb gdzie „a" jest kodem błędu, natomiast „bbbbbbbb" adresem który go spowodował (patrz tabela 1). Dla przykładu: Guru Meditation #00000005.01234567 oznacza, że w tasku znajdującym się pod adresem 01234567 nastąpiła próba dzielenia przez zero.

2) błędy systemu (lub programu).

Guru spowodowane przez program mają następujący wygląd: Guru Meditation #ccddeeee.ffffffff „cc" informuje o tym przez którą część systemu został spowodowany błąd. Wartość ta jest w ścisłym związku z polem „eeee" (patrz tabelka 2). Jeżeli wartość pola „cc" zaczyna się od „0", oznacza to że błąd jest naprawialny i zazwyczaj system sam poradzi sobie z jego eliminacją. Natomiast jeżeli pojawi się „8" sprawa wygląda gorzej. Błąd jest fatalny i system nie jest w stanie sam go naprawić. Pole „dd" informuje, co było powodem błędu (przybiera wartości jak w tabeli 3). Uwaga! W nawiasach podane są słowne odpowiedniki numerów błędów dla Azteca C (definicje znajdują się w exec/alerts.h). Dla przykładu Guru Meditation 0201000A.01234567 oznacza, że program znajdujący się pod adresem 01234567 nie może otworzyć mapy bitowej z powodu braku pamięci. Na szczęście sytuacja jest do naprawienia. Natomiast Guru #81000009 oznacza, że program chce zwolnić pamięć, która jest już wolna. Tego system nie jest w stanie pojąć (czyli nienaprawialny błąd).

Na rynku istnieje program „Gomf", który dla systemu 1.3 w większości przypadków pozwala uniknąć Guru. Ostatnia, najnowsza wersja V3.0 kosztuje około 36$. Niestety nie widziałem w Polsce legalnego dystrybutora tego programu. System 2.0 posiada podobny mechanizm już wbudowany na stałe.

I to by było tyle o „Guru Meditation". Oczywiście temat nie został całkowicie wyczerpany. Artykuł ten należy traktować jako wstęp do szerokiego zagadnienia. Więcej można dowiedzieć się z literatury specjalistycznej (czasopismo Amiga Sondernheft itp.) lub wszelkiego rodzaju książek o programowaniu na Amidze (np: Aztec C Manual).

●●●●●

Ktoś może się zapytać, co to w ogóle znaczy Guru Meditation. Otóż projektanci Amigi nie zajmowali się wyłącznie komputerami. W wolnych chwilach wymyślali różne inne ciekawe rzeczy. Przykładem może być mata "Guru Meditation", kontrolująca napięcie mięśni i służąca do odpoczynku. Konstruktorzy stwierdzili, że po zawieszeniu się systemu, należy się chwila relaksu.

tabela 2	pole „cc"
dla błędów spowodowanych przez biblioteki:	
01 lub 81	Exec Library,
02 lub 82	Graphics Library,
03 lub 83	Layers Library,
04 lub 84	Intuition Library,
05 lub 85	Math Library,
06 lub 86	CList Library,
07 lub 87	AmigaDOS Library,
08 lub 88	RAM Handler Library,
09 lub 89	Icons Library,
0A lub 8A	Expansion Library,
dla błędów spowodowanych przez urządzenia:	
10	Audio Device,
11	Console Device,
12	GamePort Device,
13	Keyboard Device,
14	Trackdisk Device,
15	Timer Device,
dla błędów spowodowanych przez porty:	
20	CIA Resource,
21	Disk Resource,
22	Misc Resource,
dla innych błędów:	
30	BootsTrap,
31	Workbencha,
32	Disk Copy.

tabela 3	pole „dd"
01	Brak pamięci,
02	Błąd tworzenia biblioteki (Create Library),
03	Błąd otwierania biblioteki (Open Library),
04	Błąd otwierania urządzenia (Open Device),
05	Błąd otwierania portu (Open Resource),
06	Błąd wejścia/wyjścia (I/O),
07	Brak sygnału.

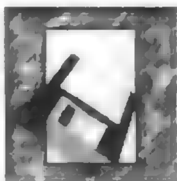
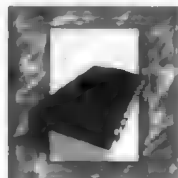


tabela 4			pola „ccddeeee"
cc	dd	eeee	
Exec Library: <AN_ExecLib>			
81	00	0001	błąd sumy kontrolnej wektora stanu wyjątkowego mikroprocesora <AN_ExecVect>,
81	00	0002	błąd sumy kontrolnej ExecBase <AN_BaseChkSum>,
81	00	0003	błąd sumy kontrolnej biblioteki <AN_LibChkSum>
81	00	0004	brak pamięci do stworzenia biblioteki <AN_LibMem>,
81	00	0005	zniszczona lista pamięci <AN_MemCorrupt>,
81	00	0006	brak pamięci do obsługi przerwania <AN_IntrMem>,
81	00	0007	wartość wskaźnika poza zakresem <AN_InitAPtr>,
81	00	0009	próba zwolnienia wolnej pamięci <AN_FreeTwice>,
Graphics Library: <AN_GraphicsLib>			
82	01	0001	brak pamięci dla listy Coppera,
82	01	0002	brak pamięci dla listy instrukcji Coppera,
82	00	0003	lista Coppera przepełniona,
82	00	0004	błąd w instrukcjach dla Coppera,
82	01	0005	brak pamięci na początek listy Coppera,
82	01	0006	brak pamięci na długą klatkę <AN_LongFrame>,
82	01	0007	brak pamięci na krótką klatkę <AN_ShortFrame>,
82	01	0008	brak pamięci na zrealizowanie wypełniania,
82	01	0009	brak pamięci na raster tekstu <AN_TextTmpRas>,
82	01	000A	brak pamięci na mapę bitową <AN_BltBitMap>,
82	01	000B	region pamięci niedostępny <AN_RegionMemory>,
82	01	0030	brak pamięci na stworzenie ViewPortu <AN_MakeVPort>,
Layers Library: <AN_LayersLib>			
83	01	0001	brak pamięci dla nowej warstwy <AN_AN_LayersNoMem>,
Intuition Library: <AN_Intuition>			
84	00	0001	nieznany typ gadżetu <AN_AN_GadgetType>,
84	00	0001	nieznany gadżet <AN_AN_BadGadget>,
84	01	0002	brak pamięci na stworzenie nowego portu <AN_CreatePort>,
84	01	0003	brak pamięci na wyświetlenie menu głównego <AN_ItemAlloc>,
84	01	0004	brak pamięci na wyświetlenie menu rozwiniętego <AN_SubAlloc>,
84	01	0005	brak pamięci na wyświetlenie górnej listwy menu <AN_PlaneAlloc>,
84	00	0006	błędna pozycja górnej listwy <AN_ItemBoxTop>,
84	01	0007	brak pamięci na otworzenie nowego ekranu <AN_OpenScreen>,
84	01	0008	brak pamięci na otworzenie rastra dla nowego ekranu <AN_OpenScrnRast>,
84	00	0009	próba otwarcia nieznanego typu ekranu <AN_SysScrnType>,
84	01	000A	brak pamięci na dodanie nowego gadżetu do listy <AN_AddSWGadget>,
84	01	000B	brak pamięci na otworzenie nowego okna <AN_OpenWindow>,
84	00	000C	zły stan systemu <AN_BadState>,
84	00	000D	zła wiadomość z IDCMP <AN_BadMessage>,
84	00	000E	niezrozumiała wiadomość <AN_WeirdEcho>,
84	00	000F	nie można otworzyć Console Device <AN_NoConsole>,
Dos Library: <AN_DOSLib>			
07	01	0001	brak pamięci na wykonanie startup-sequence <AN_StartMem>,
07	00	0002	nie można zakończyć tasku <AN_EndTask>,
07	00	0003	błędny plik <AN_QPktFail>,
07	00	0004	zwrócony nieoczekiwany plik <AN_AsyncPkt>,
07	00	0005	zły wolny wektor <AN_FreeVec>,
07	00	0006	błąd w kolejności bloków na dysku <AN_DiskBlkSeq>,
07	00	0007	zniszczona mapa bitowa dysku <AN_BitMap>,
07	00	0009	zła suma kontrolna na dysku <AN_BadChkSum>,
07	00	000A	błąd podczas wykonywania operacji dyskowych <AN_DiskError>,
07	00	000C	zła nakładka <AN_BadOverlay>,
Ram Library: <AN_RAMLib>			
08	00	0001	błędna lista pamięci <AN_BadSegList>,
TrackDisk Device: <AN_TrackDiskDev>			
14	00	0001	błąd podczas szukania na dyskietce <AN_TDCalibSeek>,
14	00	0002	<AN_TDDelay>,
Timer Device: <AN_TimerDev>			
15	00	0001	błędny requester <AN_TMBadReq>,
15	00	0002	<AN_TMBadSupply>,
Disk Resource: <AN_DiskRsrc>			
21	00	0001	błąd sygnału zmiany dysku <AN_DRHasDisk>,
21	00	0002	<AN_DRIntNoAct>,
Expansion Library: <AN_ExpansionLib>			
0A	00	0001	błąd dołączonego urządzenia <AN_BadExpansionFree>.



HISTORIA JEDNEGO ROZSZERZENIA

Szczecin, 29.03.92

Szanowna Redakcjo

Odpowiadając na Wasz apel chciałbym podzielić się z Wami oraz innymi czytelnikami Amigowca swoimi doświadczeniami związanymi [redacted] użytkowaniem rozszerzeń pamięci.

Otóż po stwierdzeniu, że 1 MB to nadal zbyt mało do nieco poważniejszych zastosowań niż gry, udałem się do pobliskiego sklepu z komputerami. Uwagę moją zwróciło estetyczne pudełko koloru wiśniowego z napisem EUREKA EXPANSION RAM - CARD 2 MB. Zawartość była nie mniej interesująca.

Solidnie wykonana karta zawierała szesnaście kości DRAM firmy Hitachi w podstawkach oraz zegar. Oprócz tego, rozszerzenie składało się [redacted] podstawki pod Gary`ego, trzech kabelków, dyskietki testującej i instrukcji w języku angielskim, z której dowiedziałem się, że karta umożliwia rozszerzenie pamięci naszej Amigi 500 do 1 MB Chip i 1.5 MB Fast (Slow) lub do 0.5 MB Chip i 1.8 MB Fast (Slow), a po zainstalowaniu w startupie firmowego programu "Extra" dodatkowo o 256 kB Fast (Slow).

Zdecydowałem się na zakup Eureki tym bardziej, że cena nie była wygórowana (marzec br. - 1.9 mln zł).

Po zainstalowaniu i kilkudniowej eksploatacji rozszerzenia zaobserwowałem niepokojące zjawiska, takie jak np. zbyt częste zawieszanie się programów działających dotąd poprawnie, read - write error występujący podczas odczytu danych zapisanych w ramdysku RAD: lub RRD: oraz zniekształcenia grafiki i animacji (pod DPaintem lub SpectraColor).

Wszystko wskazywało [redacted] wadliwe kości RAM, więc użyłem firmowego programu testującego, który nie wykazał żadnej usterki. Również posiadany przeze mnie wcześniej "Doctor Ami..." nie wykrył błędów pamięci.

Jedynym skutecznym programem testującym okazał się RAM - Test V3.11, napisany przez Markusa van Kampena, dołączany przez niemiecką firmę Gigatron do własnych rozszerzeń, który natychmiast [redacted] bezbłędnie zdemaskował aż SIEDEM uszkodzonych kości.

Po wymianie, dwie następne Eureki zachowywały się identycznie. Przeciętnie połowa kości RAM była „padnięta".

Dzięki uprzejmości sprzedawcy (w innym sklepie) przetestowałem również kilka rozszerzeń 0.5 MB tej samej firmy. Wyniki testu identyczne. Przeciętnie co druga kość nadawała się do wyrzucenia na śmietnik.

Podsumowując, rozszerzenia firmy Eureka są niczym innym jak tylko elegancko opakowanym bublem „made in Taiwan".

Co gorsza, bubel ten oferowany jest przez niemal wszystkie sklepy i stoiska komputerowe (z wyjątkiem firmy Handwit) w Szczecinie, co skłoniło mnie do napisania niniejszego listu.

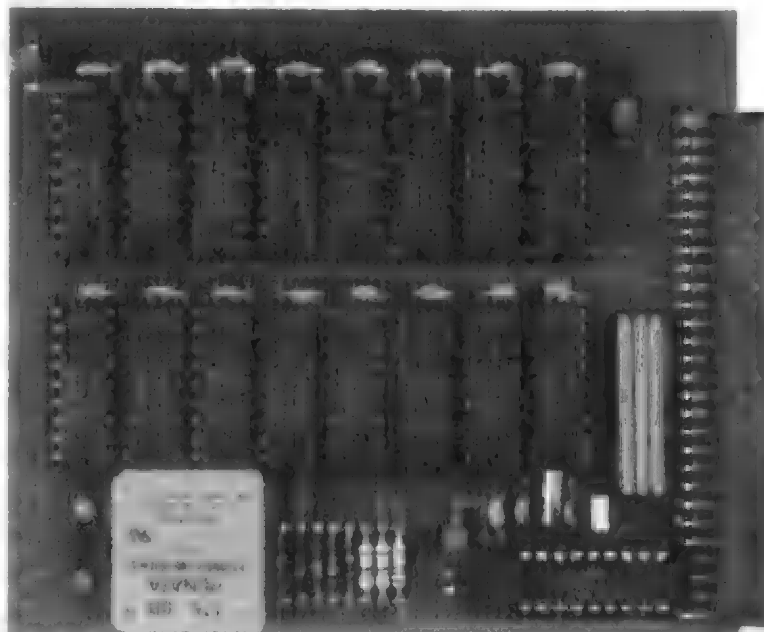
Natomiast od dwóch tygodni używam niemieckiego rozszerzenia Vector RAM - Board 2 MB A500i. Możliwości konfiguracji pamięci są takie same jak w Eurece. Oprócz tego na płytce głównej znajduje się clock - protector i wyłącznik akumulatorka. Wymiarami przypomina nieco rozszerzenia 0.5 MB z uwagi [redacted] zastosowanie czterech kości DRAM 4x1 Mbit firmy Toshiba w obudowach typu ZIP. Nie jest ono już tak elegancko opakowane i wyposażone w dyskietkę testującą, lecz [redacted] najważniejsze działa jak dotąd poprawnie zarówno zegar i pamięć. Cena powyższego rozszerzenia 2.2 - 2.5 mln zł.

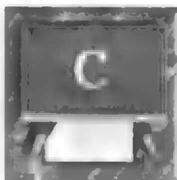
Mam nadzieję, że za pośrednictwem Amigowca zdołam ostrzec potencjalnych nabywców rozszerzeń Eureki przed ich kupnem, [redacted] tych którzy już zdążyli je nabyć przekonam do dokładnego przetestowania ich i ewentualnego zwrotu.

Na zakończenie serdecznie pozdrawiam redakcję Amigowca i życzę wielu sukcesów.

Jacek [redacted]
Szczecin

Wygląda to mniej więcej tak:





Kurs C

wejście i wyjście

Bohdan R. Rau

*/**

„Ależ tak!" - rzekł Puchatek. - „Wchodzi doskonale!"
„Ależ tak!" - rzekł Prosiaczek. - „Wychodzi doskonale!"
„Nieprawda?" - powiedział Kłapouchy. - „Wchodzi i wychodzi doskonale."

A. A. Milne, Kubuś Puchatek
(przekład Ireny Tuwim)

**/*

Najpierw trochę historii.

Pierwsze komputery (nie sięgam tu do prehistorii i maszyn typu MARK czy XYZ) miały tylko jedno urządzenie wejścia/wyjścia - dalekopis. Rozwiązanie takie miało kilka istotnych zalet: po pierwsze dalekopisy były produkowane seryjnie i adaptacja gotowego urządzenia była najprostsza z możliwych, po drugie informacje wprowadzane były typową klawiaturą maszynową i wyprowadzane normalnym drukiem, a więc nie trzeba było żadnych specjalnych kodów. Po dalekopisach pojawiły się następne urządzenia - [illegible] więc najpierw czytniki i perforatory taśmy, czyli urządzenia niejako związane z dalekopisem, potem drukarki wierszowe oraz czytniki i perforatory kart, urządzenia pamięci magnetycznej oraz monitory ekranowe. Wszystkie te urządzenia przystosowane były do przesyłania kolejnych znaków - przez długi okres monitory ekranowe działały w taki sam sposób jak dalekopisy.

Potem pojawiły się urządzenia działające na innych zasadach. Już zapis na bębnie magnetycznym pozwalał na w miarę sensowny dostęp do danych - nie trzeba było czekać [illegible] przewinięcie się taśmy - monitory ekranowe pozwalały [illegible] wyświetlanie nie tylko kolejnych wierszy tekstu, ale dowolnego znaku czy też w monitorach graficznych nawet pojedynczego piksela w dowolnym miejscu ekranu. Mimo rozwoju urządzeń zewnętrznych okazało się, że wiele zastosowań można sprowadzić właśnie do kolejnego przesyłania pojedynczych bajtów, a najwygodniejszym urządzeniem do wprowadzania bardziej skomplikowanych poleceń pozostaje pracujący w trybie dalekopisu monitor ekranowy i rozbudowa[illegible] klawiatura maszyny do pisania.

W większości języków programowania istnieje albo urządzenie o nazwie *typewriter* (np. FORTRAN) albo milcząco zakłada się, że domyślnym urządzeniem komunikacji programu z otoczeniem jest dalekopis. Przypomnijmy sobie choćby stary, dobry BASIC na C64 i działanie instrukcji INPUT i PRINT.

Choć tworzenie języków programowania szło całkiem nieźle, nie tak dobrze jednak działo się w zakresie wejścia/wyjścia. Tutaj rzeczywiście nie posunięto się daleko od FORTRANu. Niektóre języki (Algol 60) nie definiowały w ogóle tych operacji, pozostawiając je implementacji. Inne (Simula 69, Algol 68) miały zestaw wielu funkcji, z których każda drukowała coś innego. Przykładowo fragment programu w Simuli:

```
outtext("x(");
outint(i);
outtext(")-");
outfrac(x);
outimage;
```

daje dobre wyobrażenie o trudnościach programowania prostych wydruków. Lepszy pod tym względem był FORTRAN i jego formatowane wejście/wyjście. Ten sam efekt [illegible] powyżej dałby fragment programu (w wersji 1900):

```
WRITE(6,1001)I,X
1001 FORMAT
(1X,'x[',I0,']=',F0.0/)
```

Najprostszymi w użyciu były oczywiście instrukcje PRINT w językach BASIC czy FORTRAN 77. W BASICu wyglądałoby to tak:

```
[illegible] "x[";i;"]=";x
```

Tu jednak pojawiają się trudności z formatowaniem wydruku. O ile w Simuli można wywołać procedurę np. outint z dwoma parametrami z których drugi to wielkość pola zajętego przez wyprowadzaną liczbę, a w FORTRANIE zamiast swobodnego formatu I0 użyć można konkretnego (np I4) wzorca definiującego szerokość, o tyle w BASICu trzeba użyć do tego kolejnych instrukcji PRINT USING.

Język C stanowi pod tym względem rozwiązanie kompromisowe. Pozwala on na użycie typowych dla danego komputera funkcji (np. **Read** AmigaDOSu), zawiera jednak propozycje własnych funkcji standardowych, których użycie powinno być identyczne dla wszystkich komputerów. Funkcje te stanowią kompromis między podanymi rozwiązaniami. Można używać oddzielnych funkcji do wyprowadzania każdej wartości (jak [illegible] Simuli), można też używać podobnych do fortranowskich, pozwalających jednak na bardziej precyzyjne i prostsze ustalanie wyglądu wyprowadzanego tekstu formatów. I tu domyślnym urządzeniem jest tzw. konsola, czyli nic innego jak wspomniany już zestaw klawiatura/monitor symulujący pracę dalekopisu.

O konsoli trochę może później, zajmijmy się na razie wejściem i wyjściem nie tylko w C, ale w Amidze w ogóle.

Otóż - [illegible] jest bardzo ważne - każde urządzenie zewnętrzne programuje się identycznie. Tak samo więc pisać można na ekranie, drukarce czy na twardym dysku. W związku z tym istnieje kil[illegible] funkcji, które są stałe [illegible] każdego strumienia przesyłanych danych.

Wywołanie funkcji **Open** musi poprzedzać każdą (no, prawie, ale o tym potem) operację związaną z przesyłaniem danych. Ma ona dwa parametry. Pierwszy z nich to nazwa - a więc nazwa pliku na dysku (np "df0:moja/program.c"), określenie fizycznego (np. drukarka "PRT:") czy programowego (jak syntezator mowy "SPEAKER:") urządzenia czy też deklaracja konsoli (np. "CON:10/20/600/200/Okno"). Drugim z nich jest tryb dostępu. Możliwości są dwie: MODE_OLDFILE (używany przede wszystkim [illegible] już plików na dyskach) oraz MODE_NEWFILE. Funkcja [illegible] wartość, która [illegible] prawda jest nam potrzebna do dalszego działania, [illegible] w sobie nie jest interesująca, stanowi bowiem wskaźnik BCPL-u do pewnej [illegible] mądrej i bardzo nieciekawej struktury **FileHandle**. Zaznaczam, że w obu przypadkach otwarcia kanał jest dostępny zarówno dla odczytu, jak i zapisu informacji. Jeśli z jakichś powodów jedna [illegible] operacji nie będzie możliwa (trudno trochę komputerowi czytać [illegible] drukarki) wywołanie odpowiedniej funkcji nie spowoduje żadnego błędu, nie będzie miało jednak skutku.

Użycie MODE_OLDFILE powoduje odnalezienie pliku na dysku i ustawienie [illegible] pozycji początkowej. W ten sposób [illegible] czytania dostępny będzie zawsze pierwszy znak. MODE_NEWFILE powoduje skasowanie istniejącego ewentualnie pliku i utworzenie nowego [illegible] długości 0. Jeśli w czasie pisania zajdzie konieczność przedłużenia pliku, zajmuje się tym AmigaDOS, a nie my.

Jeśli z jakichś powodów system nie może udostępnić nam tego, co chcemy (brak pliku na dysku [illegible] MODE_OLDFILE, nie podłączona drukarka czy też brak handlera SPEAKER) funkcja zwróci wartość 0.

Zwróconą wartością będziemy się posługiwać przy następnych funkcjach. Będą nimi **Read** [illegible] **Write**. Każda z nich ma trzy parametry. Pierwszy z nich to zwrócona przez **Open** wartość handlera pliku, druga [illegible] adres obszaru pamięci w którym znajdują się wypisywane czy też mają znaleźć się wczytane kolejne bajty, trzecią [illegible] ilość wprowadzanych lub wyprowadzanych znaków. Obie funkcje zwracają ilość rzeczywiście przetworzonych bajtów.

Ostatnią funkcją z tego najniższego poziomu jest **Close** wywoływana z jednym tylko parametrem - handlerem. Od chwili wywołania tej funkcji program nie ma już dostępu do związanego z handlerem kanału.

Niewiele tego, prawda? Na wszelki wypadek prościutki program jako przykład (***listing #1***).

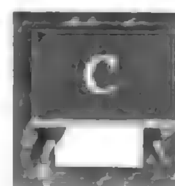
Dużo wygodniejsze jest użycie funkcji poziomu języka C. Są one następujące:

```
FILE *fopen(char *nazwa,char *tryb);
void fclose(FILE *stream);
int fgetc(FILE *stream);
int ungetc(int znak,FILE *stream);
int fputc(int znak,FILE *stream);
char *fgets(char *bufor,int wielkosc,FILE *stream);
char *fputs(char *bufor,FILE *stream);
int fread(char *bufor,int rozmiar,int ilosc,FILE *stream);
int fwrite(char *bufor,int rozmiar,int ilosc,FILE *stream);
int fscanf(FILE *stream,char *format,...);
int fprintf(FILE *stream,char *format,...);
```

Nie będę omawiał wszystkich funkcji - nie ma to być [illegible] końcu elementarz - powiem tylko o kilku trudniejszych lub mniej znanych sprawach:

FILE jest tu strumieniem danych. Z każdym z nich związany jest nie tylko otwierany kanał, [illegible] kilka istotnych informacji dotyczących sposobu zeń korzystania, zależnych zarówno od podanego drugiego parametru **fopen**, jak i natury [illegible]mego kanału. A parametrem tym nie musi być tylko "r" czy "w", możliwości jest o wiele więcej.

W systemie 5.0 napis określający tryb otwarcia składać się może z jednego do trzech zna-



ków. Oto znaczenie wszystkich kombinacji:

"r": udostępnienie pliku tekstowego do czytania,
"w": obcięcie istniejącego do zerowej długości lub utworzenie nowego pliku tekstowego oraz udostępnienie go do pisania;
"a": udostępnienie pliku tekstowego (ewentualne utworzenie) do dopisywania;
"rb": udostępnienie pliku binarnego do czytania;
"wb": obcięcie istniejącego do zerowej długości lub utworzenie nowego pliku binarnego oraz udostępnienie go do pisania;
"ab": udostępnienie pliku binarnego (ewentualne utworzenie) do dopisywania;
"r+": udostępnienie pliku tekstowego do czytania i pisania;
"w+": obcięcie istniejącego do zerowej długości lub utworzenie nowego pliku tekstowego oraz udostępnienie go do czytania i pisania;
"a+": udostępnienie pliku tekstowego (ewentualne utworzenie) do czytania i dopisywania;
"rb+" lub "r+b": udostępnienie pliku binarnego do czytania i pisania;
"wb+" lub "w+b": obcięcie istniejącego do zerowej długości lub utworzenie nowego pliku binarnego oraz udostępnienie go do czytania i pisania;
"ab+" lub "a+b": udostępnienie pliku binarnego (ewentualne utworzenie) do czytania i dopisywania;

Jeśli pierwszą literą jest 'r' a plik nie istnieje, **fopen** zwraca wartość 0.

Jeśli pierwszą literą jest 'a', wszystkie operacje pisania dokonywane są na końcu pliku niezależnie od wywoływania funkcji **fseek**.

Jeśli drugim lub trzecim znakiem jest '+', możliwe jest zarówno czytanie, jak i pisanie z następującymi ograniczeniami:

a) każda operacja czytania następująca po operacji pisania powinna być poprzedzona opróżnianiem bufora (wywołaniem funkcji **fflush**) lub pozycjonowaniem pliku (funkcje **fseek**, **fsetpos** oraz **rewind**);
b) każda operacja pisania następująca po operacji czytania powinna być poprzedzona pozycjonowaniem pliku, chyba że operacja czytania wskazała na koniec pliku.

Pojęcie pliku binarnego nie ma co prawda sensu w Amidze, gdzie pliki binarne i tekstowe traktowane są identycznie, wprowadzone zostało dla kompatybilności z systemem MS-DOS i innymi.

I tak na przykład wywołanie fopen("con:","r+") otwiera konsolę typu CON do komunikacji w obie strony. Jest to równoznaczne z udostępnieniem nowego kanału typu dalekopis.

Każdy strumień może być w pełni buforowany (tzn. każda operacja dokonuje się nie na fizycznym pliku, a na obszarze pamięci i w odpowiednim momencie następuje przestanie), buforowany liniowo (tak np. buforowany jest strumień **stdout**) lub znakowo. Buforowanie zmniejsza zarówno czas operacji wejścia/wyjścia, jak i minimalizuje możliwości kolizji programów korzystających np. z tej samej konsoli. Ma jednak również wady, o których należy pamiętać. I tak np. fragment programu kopiującego pliki.

```
printf("Kopiowanie rozpoczete..");
kopiuj();
printf("..i zakonczone\n");
```

nie da spodziewanego efektu, lecz najpierw wykona się funkcja kopiuj, a potem dopiero wypisana zostanie linia:

Kopiowanie rozpoczete....i zakonczone

Należy tu użyć funkcji opróżniającej bufor:

```
printf("Kopiowanie rozpoczete..");
fflush(stdout);
kopiuj();
printf("..i zakonczone\n");
```

Użycie funkcji **fflush** wymusza opróżnienie bufora dokonywane normalnie przy pisaniu na konsoli znaku nowej linii.

Pamiętać należy, że o ile wszystkie strumienie otwarte przez **fopen** zostają zamknięte automatycznie po zakończeniu programu, o tyle kanał otwarty przez **Open** zostaje otwarty również po jego zakończeniu!

Niestety, nie jestem w stanie podać pełnego opisu wszystkich funkcji stdio (jest tego po prostu za dużo), postaram się jednak, aby został on zamieszczony na jednym z najbliższych dysków PD.

Zajmijmy się teraz tym nieszczęsnym dalekopisem, czyli po naszemu konsolą.

Konsola Amigi jest urządzeniem dość inteligentnym. W zależności od potrzeb możemy wybrać jeden z dwóch typów konsoli: CON: - czyli taki, który automatycznie jest otwierany jako CLI oraz RAW: - urządzenie pozwalające na pełną kontrolę klawiatury.

Otwarcie konsoli jest proste. Nazwa konsoli składa się z nagłówka określającego typ oraz położenia, wymiarów i tytułu okna konsoli. Wszystkie parametry są oddzielone znakami '/'. I tak jeśli chcemy otworzyć np. konsolę typu CON: i przydzielić do niej okno o wielkości 600*100 pikseli znajdujące się na pozycji x=10 i y=30 oraz o tytule "Moja konsola" wystarczy podać nazwę "CON:10/30/600/100/Moja konsola" jako nazwę otwieranego pliku w funkcji **Open** lub **fopen**. Tryb [illegible] funkcji **Open** jest praktycznie obojętny, należy pamiętać tylko, że jeśli chcemy mieć konsolę dwukierunkową kontrolowaną przez stdio, musimy użyć parametru "r+" (dotyczy to systemu Manx 5.0 i wyższych, w systemie 3.6 dwukierunkowe otwarcie strumienia jest niemożliwe).

Konsola taka czeka za każdym razem na skompletowanie linii przez użytkownika, a następnie znak po znaku linia ta może być odczytana. W trakcie pisania możliwe jest kilka dodatkowych czynności, jak pozycjonowanie kursora czy zmiana koloru pisania. Ponieważ są one identyczne jak w konsoli RAW: podam je później.

Konsola typu RAW: jest bardziej skomplikowana.

Otóż normalnie reaguje ona natychmiast na wciśnięcie każdego klawisza. Klawisze, do których przyporządkowane są pojedyncze znaki ASCII odpowiadają pojedynczymi znakami, inne (np. funkcyjne) wysyłają całą sekwencję znaków złożoną z prowadzącego znaku '\x9b' (ALT Escape) oraz jednego lub więcej znaków zależnych od danego klawisza. Wszystkie kody podam na końcu, teraz komendy wykonywane przez konsolę (szesnastkowo, parametry w nawiasach kwadratowych oznaczają opcjonalny dziesiętny obraz liczby):

[illegible] - cofanie kursora
0a - kursor o linię w dół (LINEFEED)
0b - kursor o linię w górę
0c - czyść okno
0d - kursor do pierwszej kolumny (RETURN)
0e - normalne wyświetlanie
0f - znaki specjalne
9b [n] 40 - wstaw n spacji
9b [n] 41 - kursor n linii w górę
9b [n] 42 - kursor n linii w dół
9b [n] 43 - kursor n pozycji w prawo
9b [n] 44 - kursor n pozycji w lewo
9b [n] 45 - kursor n linii w dół do kolumny 1
9b [n] 46 - kursor n linii w dół do kolumny pierwszej
9b [n] [3b m] [illegible] - ustaw kursor w linii n w kolumnie m
9b 4a - czyść okno od kursora
9b 4b - czyść linię od kursora
9b 4c - wstaw linię
9b 4d - usuń linię
9b [n] 50 - usuń n znaków pod kursorem
9b [n] 53 - przesuń n linii w górę
9b [n] 54 - przesuń n linii w dół
9b [n] 64 - wstaw n znaków pod kursorem
9b 32 30 [illegible] - od tej chwili znak LINEFEED pisany jest jako LINEFEED RETURN
9b 32 30 [illegible] - od tej chwili znak LINEFEED pisany jest jako LINEFEED
9b [n] 74 - [illegible] wysokość konsoli w liniach
9b [n] 75 - ustaw szerokość linii w znakach
9b [n] 78 - ustaw lewy margines w pikselach
9b [n] 79 - ustaw górny margines w pikselach
(powyższe cztery komendy jeśli pominięty jest parametr ustawiają wartości domyślne)
9b 30 20 70 - kursor niewidzialny
9b 30 70 - Kursor widzialny
9b [styl] [3b kolor liter] [3b kolor tła] 6d - ustaw parametry pisania:
0 - styl normalny
1 - pogrubiony
3 - kursywa
4 - podkreślony
7 - rewers
30..37 kolor znaków 0..7
40..47 kolor tła 0..7

Konsola potrafi jednak więcej. Oprócz komend pozycjonowania kursora itp. istnieją dwie dodatkowe funkcje. Pierwszą z nich jest możliwość zapytania o wymiary okna konsoli (podane w znakach). Pytanie takie zadaje się pisząc po prostu na konsolę sekwencję znaków (szesnastkowo) 9b 20 71. Konsola odpowiada następującą sekwencją (po znaku '\x9b'):

1;1;szerokość;wysokość r

Przykładowo otrzymanie odpowiedzi 1;1;75;20 r oznacza, że w oknie konsoli mieści się dwadzieścia linii a w każdej z nich 75 znaków.

Druga możliwość to odczyt przez konsolę "wszystkiego co się rusza".

Jeśli wydamy polecenie \x9bp{ gdzie p będzie dziesiętnym obrazem liczby z zakresu 1..18 uruchamiamy dodatkową możliwość konsoli. Za każdym razem kiedy zajdzie definiowane przez liczbę zdarzenie, wysłana zostanie następująca sekwencja (oczywiście od '\x9b'):

klasa;podklasa;kod;stan;X;Y;sec;mic

(ponieważ zawsze miałem zastrzeżenia do jakości druku "Amigowca" informuję, że ostatnim znakiem jest kreska pionowa czyli znak alternatywy w alfabecie języka C).

Klasa jest tu klasą zdarzenia, kod to kod wciśniętego klawisza, stan to cała reszta informacji dodatkowych, X i Y to pozycja myszy, sec i mic to [illegible] systemowy. Pole podklasy jest nieużywane.

Bity [illegible] mają następujące znaczenia:

0x0001 Lewy shift
0x0002 Prawy shift
0x0004 CapsLock
0x0008 Control
0x0010 Lewy Alt
0x0020 Prawy Alt
0x0040 Lewy klawisz Amiga
0x0080 Prawy klawisz Amiga
0x0100 Keypad (klawiatura numeryczna)
0x0200 Powtarzany klawisz
0x0400 Nieużywany
0x0800 Aktywne okno
0x1000 Lewy klawisz myszy
0x2000 Prawy klawisz myszy
0x4000 Środkowy klawisz myszy (nieużywany)
[illegible] Pozycja myszy w oknie

Natomiast klasy zdarzenia mogą być następujące:

1 Wciśnięcie klawisza
2 Wciśnięcie klawisza myszy
3 Aktywacja okna
4 Poruszenie myszy
6 Zegar
7 Wciśnięcie gadgetu
8 Zwolnienie gadgetu
9 Zezwolenie requestera
10 Selekcja Menu
11 Zamknięcie okna
12 Zmiana rozmiaru okna
13 Odświeżenie okna
15 Wyjęcie dysku
16 Włożenie dysku

Tak więc jeśli chcemy, aby nasza konsola reagowała na zmianę rozmiaru okna, musimy wydać komendę 12{ natomiast gdy jest nam to już niepotrzebne, kasujemy tę możliwość komendą 12}.

A oto sekwencje wysyłane przy naciskaniu poszczególnych klawiszy (wszystkie od '\x9b', podaję bez SHIFT i z SHIFTem):

F1 0~ 10~
F2 1~ 11~
F10 9~ 19~
HELP ?~ ?~
Kursor góra A T~
Kursor dół B S~
Kursor lewo C [spacja]A~
Kursor prawo D [spacja]@~

I jeszcze ciekawostka: Nowsze kompilatory mają w bibliotece dwie miłe funkcje **set raw** i **set con**. Wywołuje się toto bez parametrów, a służą do tego, aby przełączyć działanie konsoli CLI przydzielonej do programu z trybu CON: na RAW: i odwrotnie.

I tu uwaga: jeśli przez **fgetc** odbieramy jakąś sekwencję, musimy odebrać ją do końca przed wyprowadzeniem jakiegokolwiek znaku na konsolę (mówię tu oczywiście o systemie 5.0).

I uwaga następna: dzisiejszy program *listing #2* może być uruchomiony tylko w systemie Manx 5.0 (lub wyższym). Przepraszam wszystkich użytkowników wersji 3.6, myślę jednak, że nawet przeczytanie kodu może dać pewne pojęcie o działaniu konsoli.

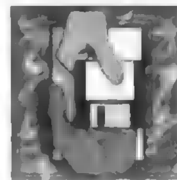
Tyle na dziś




```
/* Listing #1 */
#include <libraries/dos.h> #include <functions.h>
long FileHandler;
main() {
    int i,n;char bufor[54];
    for (i=0,n='a';i<52;i+=2,n++) {
        bufor[i]=n;
        bufor[i+1]=toupper(n);
        }
    FileHandler=Open("ram:plik_testowy",MODE_NEWFILE);
    if (!FileHandler) {
        printf("Cos mi sie nie podoba\n");
        exit(100);
        }
    Write(FileHandler,bufor,52);
    Close(FileHandler);
     FileHandler=Open("ram:plik_testowy",MODE_OLDFILE);
    if (!FileHandler) {
        printf("Cos mi sie dalej nie podoba\n");
        exit(100);
        }
     Read(FileHandler,bufor,52);bufor[52]=0;
    Close(FileHandler);
     printf("Zawartosc pliku:\n%s\n",bufor);
     printf("A teraz niespodzianka:\n");

     FileHandler=Open("ram:plik_testowy",MODE_OLDFILE);
    if (!FileHandler) {
        printf("Cos mi sie wybitnie [illegible] podoba\n");
        exit(100);
        }
    Read(FileHandler,bufor,10);
    for (i=0;i<9;i++) bufor[i]=[illegible];
    Write(FileHandler,bufor,10);
    Close(FileHandler);
     FileHandler=Open("ram:plik_testowy",MODE_OLDFILE);
    if (!FileHandler) {
        printf("Cos mi [illegible] strasznie nie podoba\n");
        exit(100);
        }
     Read(FileHandler,bufor,52);bufor[52]=0;
    Close(FileHandler);
     printf("Zawartosc pliku:\n%s\n",bufor);
     printf("Co sie zmienilo?\n");
    exit(0); }


/* listing #2 */ #include <stdio.h> #include <string.h>
  FILE *console;
 char combuffer[80]; char lastbuffer[80];
 extern int errno, struct argument {
    struct argument *next;
    short int flag;
    char *buffer;
    long int size;
    char arg[1];
    } *args;
 main() {
    int t,tc;char *c;struct argument *a;
    if (!(console=fopen("RAW:0/0/640/100/Filecopy","r+"))) exit(100);
    fprintf(console,"\2331{");
    for (;;) {
        getcommand("Kopiuj:");
        if (combuffer[0]=='!') break;
        while (c=(char *)scdir(combuffer)) {
            a=(struct argument *)malloc(sizeof(struct
argument)+strlen(c));
            if (a) {
                    a->next=args;
                    args=a;
                    strcpy(a->arg,c);
                    a->flag=0;
                    a->buffer=0;
                    a->size=0;
                    }
            }
        for (t=0,a=args;a;a=a->next)
            t+=(a->flag=conrequest("File %s",a->arg));
        if (!t) {
            fprintf(console,"Nie to nie!\n");
            goto free_args;
            }
        if (!conrequest("Kopiowac %d
plik%s?",t,(t==1)?"":t<5?"i":"ow"))
        goto free_args;
        getcommand("Kopiowac do katalogu:"); kopiowanie:
        for (t=0,tc=0,a=args;a;a=a->next) {
            if (!a->flag) continue;
            t+=readfile(a);
            if (a->buffer && a->flag) {
                    tc+=writefile(a);
                    free(a->buffer);
                    a->flag=0;
                    a->buffer=0;
                    }
            }
        if (t) goto kopiowanie;
        fprintf(console,"Skopiowano %d plikow\n",tc); free_args:
        while (a=args) {
            args=a->next;
            free(a);
            }
        }
    exit(0); }
 writefile(arg) struct argument *arg; {
    FILE *f;
    char filename[80],*cd;
    strcpy(filename,combuffer);
    if (filename[0] && filename[strlen(filename)-1]!=':')
    strcat(filename,"/");
    cd=strrchr(arg->arg,'/');
    if (!cd) cd=strrchr(arg->arg,':');
    if (!cd) cd=arg->arg;else cd++;
    strcat(filename,cd);
    fprintf(console,"Kopiuje %s do %s..",arg->arg,filename);
    f=fopen(filename,"w");
    if (!f) { err:    fprintf(console,"Nie skopiowane\nblad %d
%s\n",errno);
        free(arg->buffer);
        arg->buffer=0;
        arg->flag=0;
        return 0;
        }
    if (!fwrite(arg->buffer,arg->size,1,f)) {
        fclose(f);
        goto err;
        }
    fprintf(console,"Skopiowany\n");
    fclose(f);
    free(arg->buffer);
    arg->buffer=0;
    arg->flag=0;
    return 1; }
 readfile(arg) struct argument *arg; {
    FILE *f;
    f=fopen(arg->arg,"r");
    if (!f) {
        arg->flag=0;
        fprintf(console,"Nie moge otworzyc pliku %s\n",arg->arg);
        return 0;
        }
    if (!arg->size) {
        fseek(f,0,SEEK_END);
        arg->size=ftell(f);
        fseek(f,0,SEEK_SET);
        if (!arg->size) {
            arg->flag=0;
            fclose(f);
            return 0;
            }
        }
    if (!(arg->buffer=(char *)malloc(arg->size))) {
        fclose(f);
        return 1; /* moze w nastepnym przebiegu... */
        }
    if (!fread(arg->buffer,arg->size,1,f)) {
        free(arg->buffer);
        arg->flag=0;
        arg->buffer=0;
        fprintf(console,"Blad odczytu pliku %s\n",arg->arg);
        fclose(f);
        return 0;
        }
    fclose(f);
    return 0; }
 conrequest(fmt,arg) char *fmt;int arg; {
    int t,n;
    n=fprintf(console,fmt,&arg);
    n+=fprintf(console," t/n?");
    for (;;) {
        switch(fgetc(console)) {
            case 't':
            case 'T':fprintf(console,"\233%dD\233Ktak
\233Om\n");return 1;
            [illegible] 'n':
            case 'N':fprintf(console,"\233%dD\233K",n);return 0;
            }
        } }
 getcommand(prompt) char *prompt; {
    int ch,i,pb,ncs[10],wcon;
    strcpy(lastbuffer,combuffer);
    fprintf(console,"\2331{"); start:
    fprintf(console,"\233 q");
    combuffer[0]=0;
    fprintf(console,"%s ",prompt);
    for (pb=0;;) {
        ch=fgetc(console);
        if (ch>=' ' && ch<0x7f && pb<wcon) { [illegible]:
            fprintf(console,"\233@%c",ch);
            for (i=wcon-1;i>pb;i--) combuffer[i]=combuffer[i-1];
            combuffer[pb++]=ch;
            continue;
            }
        if (ch==0x7f) /* delete */ {
            fprintf(console,"\233P");
            for (i=pb;i<wcon-1;i++) combuffer[i]=combuffer[i+1];
            continue;
            }
        if (ch=='\b' && pb>0) /* backspace */ {
            pb--;
            fprintf(console,"\233D\233P");
            for (i=pb;i<wcon-1;i++) combuffer[i]=combuffer[i+1];
            continue;
            }
        if (ch=='\n' || ch=='\r') break;
        if (ch!=0x9b) continue;
        ch=fgetc(console);
        if (isdigit(ch)) {
            i=0; nextitem:   ncs[i]=0;
            while (isdigit(ch)) {
                    ncs[i]=(ncs[i]*10)+(ch&15);
                    ch=fgetc(console);
                    }
            while (isspace(ch)) ch=fgetc(console);
            if (ch==';') {
                    ch=fgetc(console);
                    i++;
                    goto nextitem;
                    }
            switch(ch) {
                    case '~': /* klawisz funkcyjny */
                    if (ncs[0]==2) /* klawisz F1 */ {
                        strcpy(combuffer,lastbuffer);
                        combuffer[wcon]=0;
                        [illegible] fprintf(console,"\233%dD",pb);
                        fprintf(console,"\233K%s",combuffer);
                        pb=[illegible];
                        continue;
                        }
                    continue;
                    case '|': /* zmiana wymiarow okna */
                    fprintf(console,"\x0c");goto start;
                    [illegible] 'r':
                    wcon=ncs[3]-2-strlen(prompt);
                    continue;
                    }
            continue;
            }
        if (ch=='D' && pb>0) /* <- */ {
            pb--;
            fprintf(console,"\233D");
            [illegible]inue;
            }
        if (ch=='C' && combuffer[pb]) /* -> */ {
            pb++;
            fprintf(console,"\233C");
            continue;
            }
        if (ch==' ') {
            ch=fgetc(console);
            if (ch=='A' && pb) /* << */ {
                    fprintf(console,"\233%dD",pb);
                    pb=0;
                    continue;
                    }
            if (ch=='@' && combuffer[pb]) /* >> */ {
                    for (i=pb;combuffer[i];i++);
                    fprintf(console,"\233%dC",i-pb);
                    pb=i;
                    continue;
                    }
            }
        }
    fprintf(console,"\2330 q\n"); }
```



Public Domain #6

+ Dodatek B

Tomasz [illegible]

Dzisiaj mam zaszczyt przedstawić kolejną porcję prac nadesłanych przez naszych czytelników. Ponieważ większość z autorów programów wyczerpująco opisuje ich przeznaczenie i zasady posługiwania się nimi, dlatego postanowiłem zamieścić kilka z opisów w oryginale.

Na najnowszym dysku PD znajdziecie Państwo także prawdziwą gratkę, [illegible] mianowicie emulator IBM z grafiką na Amigę!

Drugi dysk-dodatek przeznaczony jest dla wszystkich początkujących amigowców. Oprócz kilku drobnych, aczkolwiek bardzo użytecznych programów, zawiera on pełny opis dyskietki WorkBench 1.3 jaka dołączana jest do prawie każdej Amigi. A oto szczegóły...

AMIGOWY PD#6

1. Bio - Rytm v2.25 To coś (bo czy to jest użytek, czy gra ?) napisałem jako mój pierwszy poważniejszy program w Pascalu. Program ten nie jest zbyt przydatny normalnemu człowiekowi, choć jego kod źródłowy może być niejednemu pomocny do pisania własnych programów.

Po uruchomieniu program otwiera własny ekran, a na nim okno z kilkoma ważnymi (?) komunikatami. W lewym górnym rogu ekranu znajduje się gadżet zamykania okna. Kliknięcie na niego powoduje otwarcie się okna, w którym należy wybrać jedną z trzech możliwości: Wyjście, Od Początku lub Zaniechaj (okno to czasami otwiera się dopiero po chwili, np. jeśli akurat rysowane są wykresy, okno otworzy się po zakończeniu tej czynności). Po kliknięciu na ramkę otwiera się okno, w którym należy Kliknąc na ramkę nad gadżetem 'Gotowe' i wpisać nazwisko osoby, dla której [illegible] być wyznaczony biorytm. Nazwisko można podać do [illegible] znaków. Po wpisaniu nazwiska wciskamy gadżet 'Gotowe'. Otwiera się kolejne okno, w którym po kliknięciu na ramkę należy podać datę urodzenia sposobem dd-mm (czyli dzień i miesiąc dwoma cyframi, a rok od 0 do [illegible] lat). Cała data nie może przekroczyć 12 znaków. Po wpisaniu daty naciskamy RETURN. Jeśli data była podana nieprawidłowo, to ukaże się komunikat 'Źle podałeś datę !'. Należy wtedy kliknąć na gadżet 'Okay' i datę będziesz zmuszony wpisać od początku. Program wie, że luty ma 29 dni, tak że 'zagięcie' go będzie dość trudne (ale czego to ludzie nie wymyślą ?). Zresztą cały program starałem się uczynić maksymalnie 'idiotoodporny'. Wracając do programu, po prawidłowym wpisaniu daty urodzenia należy podać datę wyznaczenia biorytmu, w sposób identyczny jak powyżej. Jeśli data wyznaczenia biorytmu jest wcześniejsza od daty urodzenia, program zapyta, czy jest to prawidłowe. Jeśli nie, będziesz musiał jeszcze raz podać datę wyznaczenia biorytmu. Następnie program narysuje cztery wykresy: biorytm fizyczny, intelektualny, emocjonalny oraz ich wartość średnią. Przedział wykresu obejmuje 10 dni wstecz i 10 naprzód. Nad każdym wykresem znajduje się informacja o obecnej wartości w %. Na środku ekranu wyświetli się ramka z informacjami o człowieku, dla którego biorytm jest wyznaczany. Poniej znajdują się trzy gadżety: Nagraj Plik, Od Początku i Koniec. Działanie ostatniego jest identyczne jak gadżetu zamykania na górze ekranu. Naciśnięcie gadżetu środkowego powoduje skok do początku programu, tzn. do wpisywania nazwiska. Naciśnięcie pierwszego powoduje otwarcie się ramki, w której należy podać nazwę pliku, do którego chcemy nagrać dane. Dane obejmują: datę urodzenia i wyznaczenia biorytmu, nazwisko, liczbę dni życia oraz wartości poszczególnych biorytmów. Po wpisaniu nazwy pliku kliknij [illegible] gadżet 'Gotowe', jeśli rzeczywiście chcesz nagrać dane lub na gadżet 'Zaniechaj', jeśli nie chcesz tego robić. Jeśli wszystko będzie w porządku, nad ramką w której wpisałeś nazwę pliku ukaże się odpowiedni komunikat, a jeśli coś pójdzie nieprawidłowo, zostaniesz o tym poinformowany. Co do wydruku danych na drukarkę, to można (zapewne, gdyż nie posiadam drukarki i nie mogłem tego sprawdzić) wpisać w miejsce nazwy PRT: i posłać RETURNEM. Co do wydruku całego ekranu na drukarkę, proponuję użycie programów Graphic Dump (na dysku Workbench), programu TurboPrint lub PicSaver.

Program można uruchamiać zarówno spod WorkBencha, jak i CLI. W pierwszym przypadku otworzy się małe okno komunikacyjne. Standardowo nie przewidziałem żadnych komunikatów na tym oknie, choć jeśli zabraknie pamięci lub coś w tym stylu, komunikat zostanie właśnie tam wydrukowany.

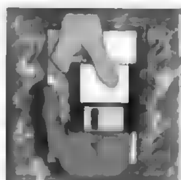
Nie zapomnij, że do swej pracy program potrzebuje mathtrans.library !

Program jest programem FreeWare, nie zgadzam się więc na dokonywanie w nim zmian czy przeróbek. Kod źródłowy dołączyłem do programu, aby pomóc tym, którzy chcieli by coś napisać w Pascalu, a nie wiedzą jak. Mogą wtedy po prostu skorzystać z mych doświadczeń lub nawet całych procedur.

Obecna wersja programu powstała 19 kwietnia 1992 roku.

Autorem programu i opisu do niego jest Kamil Iskra.

2. DESTROY - życie stało się prostsze! Jest to bardzo krótki, napisany w asemblerze program, który powoduje, że po resecie (Ctrl+A+A) zostaje wyzerowane cała pamięć RAM komputera i wszystkie znajdujące się w niej ewentualnie wirusy. Program ten powinno się umieścić w "startup-sequence" na każdym dysku, aby uchronić się przed inwazją wirusów. Niestety nie można go stosować jednocześnie z dyskiem



RAD:
Uwaga! ——
Program uruchomić można jedynie z poziomu CLI.

Autorem programu jest Cezary Biernacki.

3. IBeM v 1.05

I oto jest! Emulator programowy komputera IBM PC-XT na Amigę. W odróżnieniu od Transformera, który uruchamiał się tylko na 68000, ten emulator został stworzony do pracy na wielu Amigach. Jego niewątpliwą zaletą jest możliwość pracy w multitaskingu, tak że można jednocześnie uruchamiać programy IBM-owskie i Amigowskie. Wadą natomiast jest dosyć duża, w porównaniu [illegible] Transformerem, pamięciożerność. Prezentowana przeze mnie wersja odpala się na Amigach posiadających co najmniej 1MB pamięci! Każda próba uruchomienia emulatora [illegible] komputerze bez wymaganego rozszerzenia pamięci zakończy się zawieszeniem systemu.

Na dyskietce znajdują się następujące pliki:

- IBeM - emulator IBeM000 - wersja emulatora dla procesora 68000/68010
- mountlist - lista instalacyjna urządzeń

Aby móc czytać i zapisywać format IBM-a, na dyskietce z emulatorem powinien znajdować się messydisk.device lub CrossDos.

[illegible] naszym dysku znajduje się program messydisk.device w wersji 1.30

Wymagania sprzętowe

– Dla wersji 020/030 Wersja [illegible] przewidziana jest [illegible] Amig z procesorem 68020/68030 (lub 68040) z przynajmniej 2 [illegible] pamięci FAST. Wersja [illegible] nie działa na 68000. Przyczyną wymogu tak dużej ilości pamięci jest to, że emulator wykorzystuje cały 1 Mb dla IBM-owskiej przestrzeni adresowej. Koprocesor matematyczny nie jest wymagany, ale ostatnie wersje emulatora mogą używać jednego do emulacji 8087. Wersja 020 nie działa z niektórymi kartami GVP.

– Dla 68000/68010 Przeznaczony jest dla użytkowników Amig z procesorem 68000 lub 68010. Ten emulator odpala się również na procesorach 68020/68030, ale wersja wykonana specjalnie dla tych procesorów jest bardziej optymalna i działa szybciej. Jeżeli posiadasz kartę GVP, która uniemożliwia uruchomienie wersji dla procesorów 020/030, możesz tylko używać wersji 68000. Ta wersja nie wymaga tyle pamięci do uruchomienia co wersja 020/030 - wystarczy 1 Mb, ale do pracy przydałoby się znacznie więcej. Np. po uruchomieniu kompilatora języka Turbo Pascal 5.5 nie wystarcza już pamięci [illegible] skompilowanie krótkiego nawet programu. Użytkownicy Amigi 3000, którzy posiadają tylko 1 Mb fastu i 1 [illegible] chip powinni uruchomić właśnie tę wersję.

EMULACJA

Generalnie emulator rozpoznaje wszystkie instrukcje procesora [illegible]. Wprowadzono również emulację karty CGA. Mamy więc w trybie tekstowym do dyspozycji 4 ekrany po 0 kolumn lub [illegible] ekranów po 40 kolumn tekstu, czterokolorową grafikę w trybie niskiej rozdzielczości i dwukolorową w trybie wysokiej rozdzielczości. Niestety nie zaemulowano IBMowskiego trybu FLASH (miganie napisów itp.), a zastąpiono go podświetleniem tekstu.

Nie ma dźwięku, ani możliwości obsługi urządzeń zewnętrznych podłączanych do serial portu.

EMULATOR [illegible] stacja dysków

W celu przedefiniowania stacji dysków z Amigowskiej na IBMowską powinieneś stworzyć tzw. mountlist, czyli ciąg instrukcji określających szczegółowo parametry napędu. Minimalnie można wykorzystać jeden napęd, Maksymalnie - dwa napędy dysków elastycznych oraz jeden dysk twardy. Napędy muszą być nazwane: IBMA:, IBMB: - dla napędów dysków elastycznych oraz IBMC: - dla twardego dysku. A oto wykaz parametrów służących do zdefiniowania napędów IBMowskich:

BlocksPerTrack
Surfaces
HighCyl
LowCyl
StartCyl
DosType
Device
Unit
Flags

Owe parametry (z wyjątkiem Flags) są niezbędne i żadnego z nich nie można pominąć. Wszystkie inne ewentualne komendy są ignorowane przez emulator. Pole DosType określa rodzaj napędu i może przyjmować następujące wartości:

DosType = [illegible] 5.25" 360K
2 5.25" 1.2M
3 3.5" 720K
4 3.5" 1.44M

Normalnie Flags zostaje pominięte, ale może przyjmować wartość 1 dla napędu 5.25".

Nie wymagaj aby stacje dysków w Twojej Amidze czytały format 1.44. Jest to możliwe tylko wtedy, jeżeli posiadasz kontroler IBMa w swojej maszynie i przypiszesz jemu obsługę napędów.

URUCHOMIENIE EMULATORA

Aby wystartować emulator należy uprzednio zamontować wszystkie napędy dyskowe jakie chce się wykorzystywać podczas pracy (uprzednio jednak należy upewnić się, czy aby na pewno posiadamy wymagane rozszerzenie pamięci w swoim komputerze). Następnie wystarczy uruchomić sam emulator (IBeM). W chwilę po wykonaniu tej czynności na ekranie monitora powinny zostać wyświetlone wszystkie zamontowane urządzenia ("Using device IBMx:"). Nastąpi to w przypadku poprawnego wykonania mountlistu z poziomu CLI. Jeżeli wystąpi błąd w mountliście lub na [illegible] nie włożymy dyskietki z MS-DOSem, na ekranie ukaże się napis: "No bootable disk's found". Ponieważ każdorazowe montowanie napędów dyskowych jest dosyć uciążliwą czynnością, wykona to [illegible] Was "run-ibem" (coś w rodzaju sekwencji instrukcji wykonywanych przed uruchomieniem emulatora). Wyjście z emulatora umożliwia kombinacja klawiszy ctrl-alt-del.

Podczas uruchamiania programu istnieje możliwość zdefiniowania ilości pamięci jaką chcemy wykorzystywać. Mieści się ona w zagresie od 64 do 704 KB (np. pisząc IBeM000 640 zaemulujesz IBMa [illegible] standardową pamięcią 640KB).

Nasz plik run-ibem wygląda następująco:

failat 11 type txt mount IBMA: from MountList run ibem000

Zapis ten oznacza, że zamontowano jeden napęd dysków elastycznych 3.5" oraz uruchomiono emulator w wersji dla Amigi z procesorem M68000. Jeżeli natomiast cudownym zbiegiem okoliczności posiadamy w swojej Amidze dysk twardy i troszeczkę więcej pamięci niż 1Mb, to możemy zdefiniować plik run-ibem w następujący sposób:

failat 11 type txt mount IBMA: from MountList mount IBMB: from MountList mount IBMC: from MountList run ibem000 640

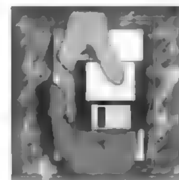
4. Konwersja v1.1

Konwersja to prosty program służący do przenoszenia plików tekstowych z polskimi znakami między standardem ks. Jana Pikula stosowanym zazwyczaj [illegible] Amidze i standardami z IBM: Mazowią, Latin 2 i DHN.

Po uruchomieniu program otwiera własne okno. Zacznijmy od opisu menu programu. Pierwsza kolumna zawiera dwa menusy: Od Autora, które z klawiatury można uzyskać kombinacją Prawa Amiga oraz [illegible]. Jego zaznaczenie powoduje wyświetlenie się kilku informacji [illegible] wersji programu, autorze itp. Wyjście, które z klawiatury uzyskasz naciskając Prawą Amigę oraz q. Tłumaczyć tego drugiego chyba nie trzeba, co najwyżej wytłumaczę się z nieco dziwnych kombinacji klawiszy. Użyłem ich dlatego, że niemal WSZYSTKIE programy używają identycznych - co [illegible] tego, że nie polskie ?

Druga kolumna zawiera również dwa menusy:

1. Zamiana RETURNów. Opcja ta jest standardowo włączona. Otóż standard ASCII na Amidze i na IBM jest nieco inny - różni się przede wszystkim kodami klawiszy używanymi do RETURNów, np. program na IBM uznałby, że tekst z Amigi jest jedną ciągłą linią ! Opcja ta usuwa to zjawisko. Powoduje ona niekopiowanie znaków [illegible] kodzie 13 i 26 (IBM RETURN i koniec pliku) oraz zamianę tekstu z Amigi na IBM. Działanie tej opcji jest równoznaczne użyciu w Dos-2-Dos Opcji -a przy kopiowaniu bądź też uruchomieniu programu Text Filter w Cross Dos.
2. Korzystanie z Bufora. Opcja standardowo włączona. Powoduje ona, że plik źródłowy zostanie najpierw wczy-



tany do pamięci, [redacted] dopiero potem nagrany do pliku docelowego. Spowalnia to nieco program, jednak jest dla posiadaczy jednej stacji bardzo pomocne (chyba że skorzystają z RAM Dysku).

Opis okna:
W oknie otwiera się 5 gadżetów: Dwa z nich to ramki podpisane z góry 'Plik Źródłowy' i 'Plik Docelowy'. Kliknięcie na nie umożliwia wpisanie nazwy pliku źródłowego i docelowego (ze ścieżką dostępu !). Plik źródłowy musi być różny od docelowego ! Obok każdego z nich znajduje się gadżet podpisany 'Standard'. Dla pliku źródłowego jest on ustawiony na 'Pikul', dla docelowego na 'Mazowia'. Gadżety te służą do poinforowania programu, w którym ze standardów jest plik źródłowy i w jakim ma być nagrany docelowy. Zmiany dokonuje się przez klinięcie na gadżet. Standardy po kolei to: Pikul (Amiga), Mazowia (IBM), Latin 2 (IBM), DHN (IBM), Bez Pol. Komentarza może wymagać ostatni 'standard': Bez Pol. Jeśli plik docelowy ustawimy [redacted] niego, to litery polskie zostaną zamienio[redacted] na angielskie, tzn. 'ą' na 'a', 'Ś' na 'S' itd. Jeszcze pewna informacja na temat Mazowii: kod 155 odpowiada u niej znakowi 'zł' - jest to jeden znak oznaczający naszą walutę. Program zamienia go w procesie konwersji na dwa znaki : 'z' i 'ł' (dla Bez Pol drugi to 'l').

Jeśli jesteś już gotów, naciśnij gadżet 'Gotowe'. W tym momencie gadżety oraz menusy z drugiej kolumny zostaną wyłączone (spróbuj je kliknąć). Przerwanie procesu konwersji jest niemożliwe, chyba że ktoś użyje 'Trzech magicznych klawiszy': Lewa Amiga + Prawa Amiga + Control. Jeśli nazwy plików źródłowego bądź docelowego będą nieprawidłowe, proces konwersji zostanie przerwany. Jeśli jednak wszystko będzie w porządku, ujrzysz komunikat: 'Proszę czekać - trwa konwersja pliku'. Jeśli w czasie konwersji nastąpi błąd (brak pamięci na bufor, błąd odczytu lub zapisu) zostanie ona przerwana i zostaniesz poinformowany [redacted] błędzie. Jeśli konwersja zostanie zakończona, ujrzysz komunikat 'Konwersja zakończona sukcesem' i jeśli była włączona opcja 'Korzystanie z Bufora' po chwili komunikat 'Nagrywanie pliku docelowego zakończone sukcesem' (lub 'Błąd podczas nagrywania pliku docelowego'). Po konwersji gadżety zostaną włączone i możesz przeprowadzić konwersję jeszcze raz lub wyjść z programu.

Program jest programem FreeWare! Nie zezwalam więc na dokonywanie w nim jakichkolwiek zmian. To samo tyczy się kodu źródłowego: możesz przy jego pomocy tworzyć własne programy, nie możesz jednak przerabiać mojego!

Obecna wersja tego programu została napisana 22 kwietnia 1992 r.

Jeśli potrzebna Ci jest konwersja na jakiś z niezamieszczonych w tym programie standardów, napisz po prostu na mój adres (który znajdziesz na dyskietce) jaki standard Ci potrzebny. Prześlij mi kody polskich liter do niego i oczywiście dysk, żebym Ci miał na co nagrać nową wersję Konwersji ! Nie obraziłbym się, gdybyś przesłał również kopertę zwrotną !

Autorem programu i opisu do niego jest Kamil Iskra, którego adres znajdziesz [redacted] dysku.

5. Test Antywirusowy Pamięci i Test Ekranu v1.11
Ten krótki program ma na celu zabezpieczenie Ciebie (a właściwie Twojego komputera) przed wirusami. Metoda jego działania jest bardzo prosta: testuje wektory Cold Capture, Cool Capture, Warm Capture i Kick Tag Ptr. Jeśli są one różne od zera ($00000000), to otwiera się okno z odpowiednim komunikatem. Po jego przeczytaniu należy kliknąc na gadżet 'Okay'. Podstawową zaletą tego protektora nad innymi jest to, że jeśli w pamięci odkryje on RAD lub RRD, okno ostrzegawcze nie otworzy się. Jest to sprawa istotna, jeśli ktoś często używa powyższych 'dysków' (większość protektorów informuje, że są to wirusy). Program nie potpafi niestety usunąć 'paskudek' z pamięci, być może któraś z późniejszych wersji będzie to już potrafiła. Po zakończeniu testu program wydrukowuje sumaryczny raport o odkrytych wirusach, wtedy też zostaniesz poinformowany, czy w pamięci znajduje się RAD lub RRD. Program działa zarówno spod WorkBencha jak i CLI (Shell). W pierwszym przypadku program otworzy własne okno, w drugim - raport zostanie wydrukowany w oknie CLI, z którego program został uruchomiony.

Program posiada również tzw. Screen Test - program testuje, czy ekran Workbencha znajduje się w standardzie NTSC (200 punktów w pionie) czy też w prawidłowym (dla niemal wszystkich Amig sprowadzanych do Polski) standardzie PAL (256). Jest to potrzebne, ponieważ z niewiadomych bliżej przyczyn Amiga czasami przełącza się na drugi standard (być może jest to spowodowane śmieciami w pamięci?).

Program jest programem FreeWare. Zgadzam się na jego kopiowanie i używanie, ale nie zgadzam się na dokonywanie w nim żadnych zmian.

Podobne zasady dotyczą dołączonego do programu kodu źródłowego. Jeśli potrafisz w istotny sposób poprawić mój program (np. usuwanie wirusów lub wykrywnie, co to za wirus znajduje się w pamięci), możesz oczywiście skorzystać z mego kodu źródłowego i zmienić go, jeśli jednak zmiany mają się ograniczyć do zmiany gadżetu 'Okay' na 'W porządku', to [redacted] tego typu zmiany nie udzielam zezwolenia.

Obecna wersja programu została napisana 6 kwietnia 1992 roku.

Jeśli zauważysz jakieś błędy w programie lub potrafisz go poprawić (np. o rzeczy wymienione powyżej), proszę o kontakt ze mną:

Autorem programu i powyższego opisu jest Kamil Iskra.

6. TechnoTom - utwór muzyczny napisany przy pomocy programu z rodziny "trackerów". Autorem kompozycji jest Tomasz Rożek.

7. Język "C" - tak, tak, to kolejny listing programu.

8. Kap-Othello - to prosta i popularna gra logiczna znana również pod nazwą "Reversi" czy "Odwracanka". Opis zasad gry jest praktycznie zbędny, po prostu uruchom i spróbuj wygrać!

AMIGOWY PUBLIC DOMAIN DODATEK "B"

POCZĄTKUJĄCY UŻYTKOWNIKU AMIGI !

Dyskietka ta jest przeznaczona właśnie dla Ciebie! Jeżeli zainteresowany jesteś nie tylko grami komputerowymi, ale pragniesz poznać co znajduje się na dyskietce o nazwie Workbench i jak korzystać z tych programów - możesz zapoznać się z zawartymi tu opisami wszystkich procedur umieszczonych na Workbenchu 1.3 i reprezentowanych przez ikony, czyli obrazki symbolizujące daną procedurę.

Mam nadzieję, że zawarte tu opisy będą wystarczająco zrozumiałe i pomogą Ci zapoznać się oraz w pełni korzystać ze wszystkich programów umieszczonych na dyskietce Workbench.

Oprócz opisu na dyskietce znajdują się również trzy programy użytkowe:
1. **SuperDuper** - program kopiujący
2. **VirusChecker 6.0** - nowa wersja programu antywirusowego
3. **SysInfo - kolejna,** nowsza wersja programu zamieszczonego niegdyś na PD#2

Szczegółowe informacje [redacted] nich znajdziesz na dysku.

Autorem opisu dyskietki WorkBench jest Andrzej Kotarski.

Prosimy o uwagi o naszej bibliotece. Piszcie do nas jakich programów poszukujecie i jakie powinny się w niej znaleźć. Czekamy również na owoce Waszej pracy.

Jak zamawiać i ile to kosztuje ?

Dyskietki PD można zamawiać przesyłając pieniądze na [redacted] konto (blankiet na stronie 13) oraz:
- podając symbol dysku PD (od #1 do [redacted] oraz "A", "B")
- podając CZYTELNIE swój adres
- podając ilość posiadanej pamięci w komputerze

Nasz dysk PD kosztuje 25 tysięcy złotych (włączone w to są koszty dysku, przesyłki, nalepki, opakowania, itp.). **UWAGA:** Dla osób prenumerujących AMIGOWCA cena jest niższa i wynosi tylko 20 tysięcy złotych !

Jeżeli ktoś sobie życzy by mu wysłać dysk PD [redacted] zaliczeniem pocztowym, to sam pokrywa koszty tego zaliczenia.

PISMA, PISEMKA

Miniony miesiąc oprócz zmian w pracy redakcji, przyniósł także większą niż zazwyczaj ilość listów. Ponieważ w wielu z nich poruszane problemy były podobne, pozwoliłem sobie na zmiany w formie zadawanych pytań. Mam nadzieję, że autorzy listów nie będą mieli mi tego za złe. Rozpocznijmy od spraw związanych ze sprzętem.

Krzysztof Nowicki

W Amidze [redacted] istnieje możliwość zainstalowania na płycie głównej dodatkowej pamięci CHIP (0.5 MB), obok już istniejącej. Jednakże po wykonaniu tej czynności i uruchomieniu komputera nalutowana pamięć jest przez niego rozpoznanwana jako FAST. Co jest tego przyczyną ?

Rozszerzenie pamięci CHIP na płycie komputera nie sprowadza się tylko do wlutowania kosci. Oprócz tego należy przełącznik oznaczony jako JP2 przestawić w drugie położenie. W tym momencie chciałbym jednak uprzedzić wszystkich użytkowników myślących, że jest to prosta metoda rozszerzenia pamięci do 1.5 MB (przy założeniu, że posiadają wkładane od spodu rozszerzenie pamięci o 0.5 MB). Zamontowanie dodatkowej pamięci CHIP powoduje, że wyżej wymienione rozszerzenia stają się bezużyteczne. Wykorzystanie dolnego slotu Amigi do dalszego rozbudowania pamięci jest możliwe tylko w przypadku tylko tych kart, które są wyposażone [redacted] tzw. przelotówkę pod GARY'ego. Powiększenie pamięci CHIP [redacted] musi być koniecznie związane z grzebaniem wewnątrz komputera. Istnieją karty pamięci (1.5 MB lub więcej), w których do nas należy decyzja, czy część tej pamięci (0.5 [redacted] ma być rozpoznawana przez naszą Amigę jako CHIP, czy też jako FAST.

Jak intrpretować wyniki programu Speeddisk V3.1 ?

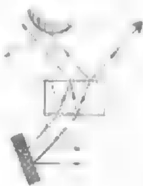
Nie jestem pewien, czy autor tego pytania podał właściwą nazwę programu. Program Speeddisk jest programem służącym do optymalizacji dyskietek i wynikiem jego pracy jest przyspieszenie niektórych operacji dyskowych. Natomiast istnieje na rynku program o nazwie DiskSpeed, który bada szybkość wszelkiego rodzaju pamięci zewnętrznych podłączonych do Amigi (ale także RAM DYSKU). Wyniki jego pracy można sobie obejrzeć w poprzednim numerze Amigowca, w artykule poświęconym dyskowi twardemu firmy GVP. Wyniki te mogą służyć do porównywania szybkości poszczególnych urządzeń zewnętrznych Amigi oraz umożliwiają dobór właściwych parametrów ich pracy (czytaj: wielkości bufora).

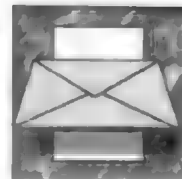
Moja stacja strasznie zgrzyta i wydaje głośne dzwięki przy przesuwie głowicy. Czego to jest objaw i czy można temu zaradzić ?

Niestety nie znam przyczyny zgrzytania stacji dysków w Amigach. Pańska stacja nie jest zjawiskiem wyjątkowym. Wszystkie Amigi, z którymi dotychczas pracowałem (mowa [redacted] Amidze 500) po pewnym czasie wykazywały opisywane objawy. Początkowo myślałem, że jest to związane z zabrudzeniem wnętrza stacji dysków. Po wyczyszczeniu jej jednak dźwięki przez nią wydawane zelżały tylko w niewielkim stopniu. Śmiem więc twierdzić, że jest to związane konstrukcją samej stacji dysków. Ponieważ jednak zgrzytanie to oprócz tego, że może stać się denerwujące, nie wpływa ujemnie na pracę samej stacji sądze, że nie jest to dla niej niebezpieczne. O uszkodzeniu mechanizmu stacji mogą jednak świadczyć pewne inne efekty dźwiękowe. Jest to bardzo głośny terkot (podobny do tego jaki wydaje stacja wczytując programy zapisane w niestandardowym formacie np. „Lemingi"). Podkreślam jednak, że są to tylko moje przypuszczenia. Jeżeli ktoś spośród Czytelników posiada więcej informacji na ten temat, chętnie udostępnimy miejsce w naszym czasopiśmie na ich opublikowanie.

O co chodzi [redacted] ray-tracingu ?

Odpowiedź znajduje się w pytaniu. Ray-tracing oznacza [redacted] polsku śledzenie promienia. Oczywiście domyślamy się, że chodzi o promień światła. Jest to wykorzystywane w programych graficznych takich jak: Imagine, Sculp, Real 3D, Caligari. Programy te różnią się od zwykłych programów malarskich tym, że operujemy w nich w przestrzeni trójwymiarowej. Na ekranie monitora widzimy trzy rzuty trójwymiarowego obiektu, który chcemy umieścić na rysunku. Oprócz tego mamy do dyspozycji ekran w którym widzimy przedmiot w perspektywie (czyli tak jak widzielibyśmy go gdyby był rzeczywisty). Obiekt ten przedstawiony jest w postaci uproszczonej jako tzw. model „druciany" - czyli wykonany z powyginanych „drutów". Jeśli przyjrzymy się jemu uważnie zauważymy, że „druty" łączą się zawsze w kształt trójkąta. Powierzchnia obiektu składa się zawsze z większej lub mniejszej ilości trójkątnych ścianek. W opisywanych programach każdemu takiemu elementowi można przypisać określone atrybuty związane z jego optyką. Są nimi współczynnik załamania, odbicia, pochłaniania, rozpraszania światła i wiele innych. Kształt obiektu oraz te atrybuty stanowią jego opis matematyczny, który wraz z takimi parametrami jak położenie obserwatora i źródeł światła, tworzy w pamięci komputera wyimaginowany trójwymiarowy świat. Mając te informacje komputer, wysyła z punktu, w którym znajduje się obserwator w kierunku źródeł światła przez wszystkie piksele na ekranie promień. Spotykając na swej drodze obiekt, w zależności od jego atrybutów, ulega odbiciu, rozszczepieniu lub załamaniu. Czasami po takim spotkaniu zamienia się w kilka promieni, które są śledzone dalej, aż do opuszczenia przestrzeni widzianej przez komputer lub trafienia na źródło światła. Po wysłaniu wszystkich promieni na powierzchni przedmiotu tworzy się mapę natężeń i koloru światła. Oczywiście proces ten jest bardziej skomplikowany, ponieważ muszą być także uwzględnione te promienie, które zostały odbite od innych przedmiotów, przez nie załamane lub rozszczepione. Mając te dane komputer jest w stanie stworzyć na ich podstawie rysunek oddający, przynajmniej w części, wygląd świata rzeczywistego. Proces tworzenia takiego obrazka trwa na





zwykłej Amidze 500 kilka do kilkudziesięciu godzin. Dlatego też wymyślono uproszczone metody tworzenia takich obrazów. Efekt jest oczywiście mniej realistyczny ale za to skraca się czas oczekiwania na wynik. Uproszczenia [redacted] polegają na tym, że promienie nie są wysyłane do każdego punktu ścianki obiektu, lecz tylko do naroży i ewentualnie środka, [redacted] parametry wyjściowe innych punktów oblicza się poprzez interpolację.

Stworzyłem przy pomocy Dpainta kilka obrazków. Czy stoi coś [redacted] przeszkodzie aby przysłać je do waszej oceny i ewentualnego wykorzystania?

Oczywiście jesteśmy zadowoleni z wszelkich przejawów twórczej działalności naszych Czytelników. Nasze czasopismo służy zresztą rozwijaniu tego typu zainteresowań wśród jego odbiorców. Jeżeli uważacie, że Wasza praca jest już na takim poziomie, aby mogła ona zainteresować szerszy krąg użytkowników możecie ją do nas przysłać. Jeśli kolegium redakcyjne uzna przysłaną pracę za godną wykorzystania w naszej pracy, postaramy się aby jej twórca otrzymał należną mu gratyfikację. Oczywiście wykorzystanie grafiki, muzyki lub gotowych programów ograniczać się będzie z oczywistych powodów, do ich zamieszczenia na dysku PD. Natomiast bardzo chętnie opublikujemy wszelkiego rodzaju teksty dotyczące Amigi, zwłaszcza zaś Waszych doświadczeń [redacted] pracy z niektórymi programami. Poszukujemy także osób, które potrafią w interesujący sposób opisać grę, podać ułatwienia do niej lub sposób jej ukończenia.

Program MultiRipper V2.1 zawie[redacted] się, gdy wczytuje program powyżej adresu $7FFFF. Czy mogę wczytać dane powyżej tego adresu, aby potem je rozpakować [redacted] pamięci powyżej $7FFFF?

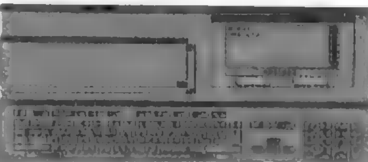
Jest [redacted] możliwe tylko w przypadku gdy Pana komputer jest wyposażony w 1 MB pamięci CHIP, albowiem od adresu $7FFFF rozpoczyna się drugie pół mega tej pamięci. Jeśli komputer nie jest wyposażony w tę pamięć, wszystko co zostanie wysłane pod ten adres, znajdzie się w próżni i komputer nie wiedząc co z tym fantem zrobić zbuntuje się. Na podstawie tego co zawarte jest w liście sądzę, że posiada Pan Amigę 500 z rozszerzeniem pamięci o 0.5 MB umieszczonym u dołu komputera. Chcąc skorzystać z zasobów tego rozszerzenia należy się odwoływać do adresu $C00000

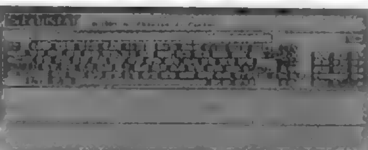
Jest wiele programików do prezentacji modułów muzycznych. Wpadłem na pomysł, by do wizualizacji odtwarzanej muzyki wykorzystać dwie diody w obudowie komputera. Jedna z nich zmieniałaby, gasła i zapalały by się [redacted] rytmie zmian dźwięku w lewym głośniku, [redacted] druga prawym. Ciekawa była by też możliwość migania diody np. zasilania [redacted] takt jednego spośród czterech kanałów dźwiękowych Amigi.

Niestety muszę autora tego pomysłu, [redacted] dokładnie pierwszego z nich, powiadomić, że istnieje już program, który wykorzystuje diody podczas odtwarzania muzyki. Nie jestem w stanie w tej chwili podać jego nazwy, znalazłem go bowiem na jakimś dysku Public Domain, a dysków takich znajduje się w redakcji ogromna ilość i nie mam dostatecznie dużej ilości czasu aby go odnaleźć. Natomiast drugi z pomysłów o ile wiem nie został jeszcze w praktyce zrealizowany, podejrzewam, że jest to spowodowa[redacted] niewielką przydatnością opisanego efektu do komponowania muzyki z użyciem komputera.

Używam programu Setkey V2.1, [redacted] on [redacted] mnie wiele wad. Często zawiesza się, gdy zmieniam mapę klawiatury. Czy jest jakiś inny program tego typu lub jakiś sposób [redacted] ominięcie tych problemów?

Rzeczywiście program, którego Pan używa nie należy do najlepszych. Na to aby uniknąć zawieszenia systemu podczas pracy z tym programem jest tylko jedna rada: zmieniając mapę klawiatury na jakimkolwiek dysku należy na ten dysk nagrać Setkey, lub na dyskietkę z Setkey'em nagrać zmienianą mapę. Jeżeli zaś chodzi o drugi z Pańskich problemów tj. brak możliwości uzyskania polskich znaków na klawiaturze po zaprogramowaniu klawiszy funkcyjnych F1 - F10 muszę stwierdzić, że nie miałem takich problemów. W swoim liście pisze Pan, [redacted] zmienia mapę klawiatury o nazwie „d-pol", a więc taką, która [redacted] już ustawione polskie znaki. Proponuję, używać do zmian oryginalnej mapy niemieckiej („d") i samemu zaprogramować rozmieszczenie liter oraz funkcji (ja sam postępowałem w ten sposób i nie miałem żadnych problemów). Jeżeli zaś chodzi o inny program realizujący te same funkcje co Setkey to mogę polecić jedynie „KeyMapEd".





Jak uzyskano [redacted] Amidze tryb „Dynamic Hires" [redacted] jakie programy z niego korzystają?

Nie znam metody uzyskiwania trybu „Dynamic Hires". Informacja taka o ile mi wiadomo nie została jeszcze nigdzie opublikowana, prawdopodobnie mają do niej dostęp tylko firmy produkujące oprogramowanie i co bardziej dociekliwi programiści (jeżeli ktoś w Polsce to rozpracował prosimy [redacted] kontakt). Jeżeli chodzi o oprogramowanie wykorzystujące ten tryb to znam tylko dwa przykłady „MACRO PAINT" oraz „DIGI VIEW". Istnieje również kilka programów potrafiących pokazać obrazek wykonany w tym trybie np. „DIRECTORY OPUS"

Czy jeśli prześlę pocztą kopię odcinka wpłaty [redacted] prenumeratę lub dysk PD to zamówienia zostanie zrealizowane szybciej?

Strasznie nad tym ubolewamy, lecz w tej chwili nie jesteśmy w stanie przyspieszyć realizacji prenumeraty na dyski PD lub nasze czasopismo. W tej chwili wysyłamy zamówione egzemplarze w terminie trzech - czterech dni od daty wydrukowania. Jest to dla nas dość uciążliwe zważywszy na fakt, że zbliżają się wakacje i zespół redakcyjny występuje często w osłabionym składzie.

Kiedy będzie można opłacić prenumeratę [redacted] następne pół roku?

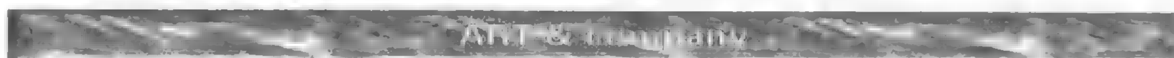
Wpłat na kolejne sześć numerów Amigowca można dokonywać już teraz.

Kończąc, chciałbym jeszcze ustosunkować się do próśb wielu osób dotyczących prowadzenia kursów programowania w naszym piśmie. Jest to dla [redacted] olbrzymi problem albowiem nie wiadomo jaki język wybrać. Jasnym jest, że nie możemy prowadzić nauki wszystkich języków, bo Amigowiec musiałby chyba liczyć z 500 stron (czego sobie życzymy). Z tego faktu wynika także drugie ograniczenie, [redacted] mianowicie przekazana w ten sposób wiedza o programowaniu mogłaby być niekompletna. Na dzień dzisiejszy rozwiązaniem są artykuły problemowe tzn. ktoś spośród Was ma konkretny problem do rozwiązania i uważa, że zainteresuje on innych Czytelników może przysłać go nam jako temat do artykułu. Oczywiście proces ten jest dwukierunkowy tzn. jeżeli ktoś z Was rozwiązał jakiś problem programistyczny to może to rozwiązanie zamieścić w naszym piśmie (prosilibyśmy jednak, aby były to przesyłki w formie gotowych artykułów, które moglibyśmy po ewentualnych konsultacjach oddawać do druku).

W ostatnim zdaniu tego artykułu po [redacted] kolejny zachęcam wszystkich do pisania listów oraz życzę sukcesów w pracy z komputerem.



RTYPERY cz. 2



Czytając poprzedni numer Amigowca (nie czytaliście?! - przeczytajcie!) dowiedzieliście się coś nie coś [redacted] grze R-TYPE oraz jej założeniach. Nawiązując do tego tematu postaramy się w tym odcinku przedstawić Wam następców (lepszych lub gorszych) owej gry.

R-TYPERY cz.[redacted]
Ogólnie rzecz biorąc, gry tego typ(er)u mogą być bardzo różne a zarazem bardzo podobne. Może zostać zachowany [redacted] schemat, [redacted] może być ściągnięty cały pomysł "żywcam". W tym odcinku jako drobne przykłady przedstawimy Wam: Z-OUT, ATOMIC ROBO-KID oraz jedną z nowszych, jednocześnie wspaniałych gier: PROJECT X !

Gra **Z-OUT** ma już ze 2 latka, a ciągle jest popularna w kręgach nałogowych graczy - i nic dziwnego: muzyka i grafika zostały zrobione [redacted] wysokim pozimie, a i sam jej pomysł wzięty został z "nieśmiertelnego" R-TYPA.

Ideą gry jest zniszczenie złego imperium [redacted] odległej planecie. Jesteś najemnym wojownikiem, dlatego też decydujesz się dla chwały i pieniędzy (a może odwrotnie?) pokonać [redacted] plansz ostrej łamaniny joysticka i nerwów, by następnie rozprawienia się z finałowym Bossem (zawsze jakiś znajdzie się [redacted] końcu każdej planszy). Autorzy gry pokusili się nawet [redacted] wprowadzenie "obcych" do gry (chyba znacie ten film). Wiadomo - wszystko zależy [redacted] wyobraźni autorów.

Sterowanie grą jest bardzo podobne jak w R-TYPIE. Tak więc dysponujesz wiązką promieni, pomocnikami, a w trakcie gry możecie skompletować różne rodzaje broni (niestety nie tak dobre jak w w/w pierwowzorze). Za to wykorzystany jest świetny pomysł, a mianowicie po wciśnięciu klawisza A dysponujecie autofirem działającym bez trzymania strzału - oszczędzi [redacted] wam palpitycznego machania palców i oczywiście joysticka.

Niektórzy twierdzą, że Z-OUT jest o wiele lepszy od R-TYPA - jest to jednak ocena subiektywna. W naszym jednak mniemaniu różnicy żadnej nie widzimy!

Dla leniwych: klawisze J-K wciśnięte na raz w czasie gry dają Wam nietykalność (pewne-sprawdziłem-ART), a kombinacje J-1,2,3,4,5,6 to przenoszenie na poszczególne plansze, to samo na klawiaturze numerycznej (z tym, że od 1 do 3 i nie w każdym przypadku) dając [redacted] przenoszenie na poszczególne etapy planszy - o ile takie są ... I to by było na tyle...noo, może jeszcze to, że gra jest na jednym dysku (to [redacted] oszczędnych).



Teraz dla odmiany przedstawiamy wam grę **ATOMIC ROBO-KID** firmy Activision. Pomysł, aby zniszczyć wrogie siły i uratować cywilizację jest ten sam, ale [redacted] została nad wyraz rozwinięta.

Otóż jesteście czymś w rodzaju mechanicznego [redacted] i wcale nie lecicie wraz z przesówem ekranu. To od was zależy, gdzie i po [redacted] lecicie, [redacted] oprócz tego plansze pną się również w górę i w dół zamiast ciągnąć się pasmem z lewej na prawo. Ciekawy jest również system wybierania i kompletowania broni. Gdy weźmiemy jedną z [redacted] możliwych, jej ikona ukazuje się na dole ekranu. Jeśli skompletujecie wszystkie - zmieniacie je wtedy przez dłuższe przytrzymanie fire'a (fajne, co?!). Również od Was zależy jaką broń weźmiecie, chcecie wiedzieć jak - zagrajcie w ATOMIC ROBO-KID !

Ogólnie o grze: jest 2 dyskowa, składa się z [redacted] leweli po 4 plansze. Ostatnie z plansz to rozprawa z jakimś stworem, ostrzegamy jednak, że jest to trudny orzech do zgryzienia dlatego podajemy cheat: [redacted] tytułowym screenie (obrazku) trzeba wpisać: TUESDAY 14TH (nie zapomnijcie o spacji w środku!). Jeśli wpiszecie go prawidłowo to po naciśnięciu fire'a ukaże się wam cała gama ułatwień. Jeśli nie - zresetujcie komputer i czynność powtarzajcie do skutku. Zapewniamy - warto !

PROJECT X - jest pięknym przykładem na udoskonaloną wersję R-TYPA. Wyprodukowana została w 1992 roku przez firmę TEAM17 (mówi wam to coś?). Muzyka i grafika jest dosłownie REWELACYJNA (dygitalizacje, syntezy mowy), ale [redacted] trzeba przeżyć na własnej skórze ! Gra chodzi na 1 MEG (jeśli go nie macie, komputer Was wyśmieje) i składa się z [redacted] dysków.

Wasza misja w grze jest prosta: dolecieć do planety RYXX, zniszczyć ruch oporu [redacted] powierzchni (to już druga plansza), splądrować jaskinie i na końcu wysadzić planetę - proste, co?

Opcji w grze jest wiele. Tak więc przyjemność ich poznania pozostawiamy wam. Natomiast trzeba wspomnieć o samym przebiegu gry. Będąc jednym z trzech możliwych statków musicie kompletować literki P (power). Na dole ekranu macie zestaw broni, którą możecie wykorzystać dzięki w/w literkom. Dzieje się to w ten sposób, [redacted] każdorazowo zaaplikowana literka pozwala użyć broni na dole, [redacted] aktywizujemy ją przez spację. I tak jeśli weźmiecie 4 literki P bez aktywizowania broni w międzyczasie, to możecie wykorzystać broń 4 od lewej (mamy nadzieję, że wytłumaczyliśmy to raczej przystępnie).

Polecamy [redacted] wybrać model CRUX II - BATTLE CLASS i skompletować SPEED, PLASMA, SIDE, MISSILE. Będziecie wtedy mieli największe szansę przetrwania ponieważ, gra jest bardzo trudna. Jak na razie cheatów do tej gry nie mamy - [redacted] szkoda!

Tak więc przedstawiliśmy wam 3 gry typu R-TYPE. Mamy nadzieję, że staną się Waszym konikiem. Mimo wszystko to przecież świetna rozrywka, przy której wyzbędziecie się swych kompleksów, albo zakompleksicie się zupełnie !

Jak dobrze pójdzie spotkamy się [redacted] miesiąc z nową serią R-TYPERÓW - a może chcecie abyśmy przeszli do innego gatunku gier? Jeśli tak, to piszcie do nas, [redacted] my z coś już dla Was znajdziemy.



Triki czerwiec'92

Jakub Sandecki

CARDIAXX
Podczas gry wciśnij klawisz P i wpisz hasło CAROLILY. Da Ci to nieskończoną energię. Możesz też wpisać imię jako RACHEL i wcisnąć SHIFT. Potem podczas spauzowanej gry wciśnij klawisz Y, by zatrzymać upływ czasu.

BARBARIAN 2
W czasie gry wciśnij kolejno: HELP (przytrzymaj), M i E. Efekt tej mieszanki klawiszowej to nieskończona energia.

FINAL FIGHT
Wciśnij klawisz P i wpisz SHERIFF FATMAN. Obaj gracze będą nieśmiertelni, a czas stanie sobie w miejscu.

ROBO COP 3
Trzymając prawy przysisk myszy wpisz THE DIDDYMEN.

FANTASY WORLD DIZZY
Wpisz swoje imię jako IMMORTAL.

LEANDER
Na ekranie z różnymi opcjami do wyboru, wpisz LTUS. Teraz, gdy spauzujesz grę klawiszem F8, F6 przeskakuje do następnego levela, a F1 - F5 wybiera rodzaje broni. Będziesz miał też nieskończone życie.

DARKMAN
W trakcie gry wpisz MEACULTA.

GODS
Hasło (password) to SORCERY.

THE GODFATHER
Spauzuj grę i wciśnij HELP. Wpisz PIZZA HUT.

FINAL BLOW
Spróbuj wcisnąć sześć razy klawisz F10 podczas walki.

ROLLING RONNY
Wpisz imię jako CHEAT w tabeli najlepszych wyników.

OH NO! MORE LEMMINGS
Dalsze kody do leveli:
1. GAHRVFLBFF
2. IIRVNLFCFG
3. MPTNHGADFM
4. RVNLGIMEFN
5. VNLGEITFFI
6. NLGMITVGFR
7. LGAMVVNHFM
8. GMMTVNLIFH
9. GAIRWNLJFH
10. IIRWNLGKFQ
11. MRWNLFALFM
12. RWNLFIMMFF
13. WNLFAIVNFO
14. NLFIIVWOFH
15. LFAMVWNPFE
16. GIMVUNLQFM
17. GAIRVLMBGO
18. IIRVLMFCGG
19. LRVDMGADGL
20. RVLMFIMEGM

RAMBO 3
Ugraj tyle punktów, byś mógł wpisać się do tabeli naj.wyników i wpisz jako swoje imię słowo RENEGADE. Odtąd klawiszami 1,2 i 3 możesz przeskakiwać pomiędzy levelami.

RETURN OF THE JEDI
Na tabeli naj.wyników wpisz DARTH VADER. Klawiszem F2 przeskakujesz levele.

STARGLIDER
Wciśnij klawisz BACKSPACE (ze strzałką, nad Return'em) i napisz JS ARG S.

STARGLIDER 2
Wciśnij BACKSPACE i wpisz WE'RE ON A MISSION FROM GOD. Wciśnij 1. K daje wszystkie możliwe bronie.

STORMTROOPER
Wpisz JAMES CAMERON na tablicy naj.wyników. Teraz wciskaj F9, gdy będziesz potrzebował nowych żyć.

SWORD OF SODAN
Ugraj dużo punktów i wpisz na tabelę naj.wyników hasełko NANCY.

THUNDERBIRDS
Kody do poziomów:
- 2 - RECOVERY
- 3 - ALOYSIUS
- 4 - ANDERSON

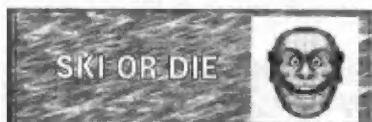
THUNDER BLADE
Na tablicy naj.wyników wpisz CRASH, klawisz HELP przenosi pomiędzy levelami.

JAGUAR XJ220



Każdy miłośnik motoryzacji z pewnością wie o wielkiej konkurencji pomiędzy firmami produkującymi samochody. Projektanci i producenci nowych modeli aut starają się różnymi sposobami zapewnić sobie prymat na rynku motoryzacyjnym. Najlepiej obrazuje to zjawisko walka toczona przez firmy produkujące samochody sportowe. Uciekają się one do coraz dziwniejszych sposobów reklamowania swoich produktów. Na bardzo ciekawy pomysł wpadła brytyjska firma Jaguar, która postanowiła reklamować swoje najnowsze cacko - JAGUAR'a XJ220 (kosztującego „tylko" 400,000 funtów) nie tylko tradycyjnymi metodami takimi jak np. modele firmy Matchbox, ale również na ekranach komputerów. W tym celu koncern ten rozpoczął współprace z firmą softwarową CORE DESIGN, znaną z takich hitów jak Thunderhawk czy Wolfchild. Efektem tej współpracy jest gra pt. JAGUAR XJ220. Jak sama nazwa wskazuje jest to nowy wyścig samochodowy bardzo przypominający słynnego LOTUS'a. Bierzemy udział w wyścigach, gdzie zmagamy się z przeciwnikami zasiadającymi za kierownicami takich pojazdów jak: Porsche, Ferrari, Lamborghini (każdy z tych pojazdów jest łatwy do rozpoznania na ekranie). Rajdy odbywają się w dwunastu różnych krajach na ponad 36 torach. Podczas jazdy napotykamy na różne utrudnienia. Warunki pogodowe ulegają zmianie. Mgła, deszcz, śnieżyca, czy też burza piaskowa utrudnia nam jazdę. Po drodze przejeżdżamy przez liczne mosty, tunele, wzniesienia. Wszystko to wpływa na urozmaicenie gry. Grafika jest bardzo ciekawa i różnorodna. Każdy etap wiedzie przez zróżnicowane krajobrazowo tereny. Poszczególne elementy graficzne są bardzo czytelne. Ruch naszego pojazdu jest płynny. Poza samą jazdą musimy również kupować i wymieniać części, które ulegają zużyciu podczas rajdu. Ta część gry zasługuje na szczególną uwagę. Nasz samochód jest pokazany w kilku rzutach i wymieniając poszczególne części widzimy gdzie są one umieszczane. W grze może uczestniczyć jeden lub dwóch graczy, wtedy ekran jest podzielony na dwie części. Program posiada również opcję projektowania tras, co jest nowym elementem w tego typu grach i wpływa na jeszcze większe ich urozmaicenie. Mocną stroną programu jest dźwięk. Tak jak w Lotusie możemy wybrać sobie melodyjkę, która będzie nam towarzyszyła na danym etapie. Poszczególne utworki nie są jednak tak dobre jak w Lotusie. Ogólnie JAGUAR XJ220 to bardzo dobry program w pełni wykorzystujący walory wcześniejszych gier tego typu i wnoszący nowe elementy. Myślę, że „zapaleni kierowcy" będą nim usatysfakcjonowani. Zajmuje on dwa dyski i działa tylko z rozszerzeniem pamięci (co najmniej 1MB).





JAGUAR XJ220
CORE DESIGN

SKI OR DIE

Dużą popularnością w świecie gier komputerowych cieszą się gry sportowe. Firmy softwarowe zarzucają rynek programami komputerowymi symulującymi różnorodne dyscypliny sportowe zarówno letnie, jak i zimowe. Oprócz standardowych konkurencji pojawiają się często różne „nietypowe". Są to bardzo zabawne próby, którym poddawany jest człowiek. Do tego typu gier zaliczyć można takie programy jak: California Games, Bobo czy też Viz. Do tej właśnie grupy należy SKI OR DIE. Jest to zbiór kilku rzadko spotykanych dyscyplin. Są to:

• Snowboard half-pipe czyli jazda po śniegu na desce, podczas której należy wykonywać różne ewolucje. Gdy znajdujemy się na prostej to możemy przeskakiwać nad przeszkodami, łapać deskę w jedną rękę, a drugą podnosić do góry oraz wykonywać obróty. Gdy znajdujemy się na stoku możemy jechać na krawędzi toru, skakać w górę z ugiętymi nogami oraz stawać na rękach na krawędzi stoku. Podwójna górka umożliwia nam zrobienie efektownego salta. Wszystko to jest odpowiednio punktowane i decyduje o zajętym przez nas miejscu.

• Innertube trash - taki wyścig gumowych pontonów po muldach.

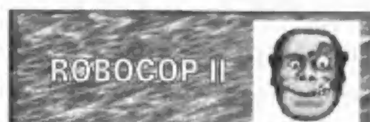
• Acro aerials - są to skoki akrobatyczne na nartach. Możemy wykonywać salta, śruby oraz szpagat),

• Downhill blitz czyli zjazd po muldach podczas którego można wykonywać akrobacje.

• Snowball blast - pojedynek na śnieżne kulki z bandą nieznośnych dzieciaków.

Grze towarzyszą różne wesołe wydarzenia. Grafika jest ciekawa i przejrzysta, a efekty dźwiękowe „dopasowane" do akcji. Możemy rozgrywać zawody w poszczególnych dyscyplinach lub też we wszystkich w ramach turnieju (wówczas o zwycięstwie decyduje suma punktów z pojedyńczych konkurencji). Gra zajmuje jeden dysk i nie wymaga rozszerzenia pamięci.





SKI OR DIE
ELECTRONIC ARTS

ROBOCOP II

Przywykliśmy już do faktu, że każdy kasowy (nie oznacza to, że dobry) film staje się scenariuszem gry komputerowej. W produkcji tego typu oprogramowania specjalizuje się firma OCEAN. Zwykle są to niesamowite kicze, zawsze jednak posiadające dobrą oprawę muzyczną i graficzną. W przypadku gry ROBOCOP III nie jest to prawdą. Program ten ma może stać się prawdziwym hitem. Jest to pewnego rodzaju nowa era w dziedzinie gier przygodowych i zręcznościowych opartych tylko na grafice wektorowej. Właśnie to wyróżnia ten program. Doskonała jest bardzo szybka i płynna animacja. Gra dzieli się na dwie części zręcznościową i przygodową. W pierwszej mamy do wyboru: jazdę samochodem w pościgu za skradzionymi autami, walki uliczne (na ulicach miasta szaleją wzajemnie zwalczające się gangi, nasz Superglina ma przerwać tę wojnę), uwalnianie zakładników znajdujących się w rękach porywaczy (chodzimy po wieżowcu wykańczając terrorystów), latanie Gyropackiem (najnowszym pojazdem policyjnym) oraz walki wręcz (konkurencja OCP stworzyła robota-



ninje, tylko Robocop ma szansę go pokonać, jest to coś w rodzaju trójwymiarowego karate). Druga część tej gry to połączenie wszystkich elementów zręcznościowych i obrona prawa w Detroit. Podczas gry możemy „patrzeć" z różnych punktów np. przez Robocopa, lub za pomocą jednej z zewnętrznych kamer. Gra wprowadza nas we wspaniałą atmosferę. Czuje się akcję, podniecenie, potęgę, którą władamy poprzez naszego bohatera. Niemały wpływ na taką atmosferę ma dźwięk. Uważam, że jest to naprawdę fenomenalny program, który już niedługo stanie się pierwowzorem dla innych gier. Zajmuje on trzy dyski, co przy tej grafice nie wydaje się dużą liczbą oraz wymaga przynajmniej 1MB pamięci.

SUPER HIPER

GRAFIKA
DŹWIĘK
MOTYWACJA
POMYSŁ
OGÓLNIE

ROBOCOP II
OCEAN

YOGI'S BIG CLEAN-UP

„Wielkie sprzątanie misia Yogi" to znakomita zabawa dla najmłodszych. Jest to bajecznie kolorowa platformówka w której nasze zadanie polega na sprzątaniu parku Jellystone. Wcielamy się w postać ulubieńca dzieci, bohatera kreskówek, misia Yogi. Nasz pupil pragnie udowodnić, że nie jest takim zwyczajnym misiem. Postanawia oczyścić swój rezerwat z różnych „śmieci" pozostawionych przez nieuważnych turystów. Chodzi więc po parku w poszukiwaniu ośmiu przedmiotów znajdujących się w różnych zakamarkach. Oprócz tego Yogi'ego czekają jeszcze cztery inne zadania. Gdy wypełni on wszystkie swoje misje znów stanie się ulubieńcem kierownictwa rezerwatu. Przez cały czas nasz bohater jest niepokojony przez wrogo nastawione ryby, żaby, wrony i różne inne leśne istoty. Teren po którym porusza się nasz miś jest wspaniale dostosowany do potrzeb i możliwości najmłodszych użytkowników Amigi. Postacie i przedmioty są duże, łatwe do rozpoznania. Animacja jest płynna. Uważam, że jest to idealna gra dla dzieci. Zajmuje ona jeden dysk i nie wymaga rozbudowy pamięci.

DOBRY PLUS

GRAFIKA
DŹWIĘK
MOTYWACJA
POMYSŁ
OGÓLNIE

YOGI'S BIG CLEAN UP
HI TECH

OVER THE NET

Zbliża się lato - pora wakacji i wesołych zabaw. Wielu z nas wyjedzie nad morze. Czekają nas tam liczne atrakcje. Morskie kąpiele i wesołe igraszki na plaży. Lato niestety szybko mija i pozostają tylko miłe wspomnienia. Firma Genias podjęła próbę przeniesienia cząstki tych wspaniałych wrażeń na ekrany naszych komputerów. Tak powstała gra OVER THE NET - plażowa siatkówka, bardzo popularna na zachodzie forma letniej rozrywki. Tylko w ogólnych zarysach przypomina ona dyscyplinę sportową o tej samej nazwie. W grze biorą udział dwie drużyny po dwóch zawodników w każdej. Reszta zasad pozostaje bez zmian. Program oferuje nam możliwość rozgrywania pojedyńczych spotkań oraz udział w Pucharze Morza, który odbywa się na cudownych plażach w tak malowniczych zakątkach świata jak np. wyspy Fidżi, Miami, Ibiza czy też Rimini. Po każdym spotkaniu pojawia się plansza, która informuje nas o liczbie zdobytych przez drużynę punktów oraz pokazuje nam ilość bloków ścięć wykonanych przez poszczególnych zawodników. Możemy decydować o liczbie setów w meczu i o tym, czy gramy z punktem przy zmianie serwisu czy też bez. W grze może brać udział maksymalnie czterech graczy (potrzebny jest wtedy specjalny interface). Grafika i dźwięk stoją na bardzo wysokim poziomie. Jest to bardzo dobry program, który spodoba się licznemu gronu miłośników gier komputerowych. Zajmuje on jeden dysk i nie wymaga rozszerzenia pamięci.

SUPER

GRAFIKA
DŹWIĘK
MOTYWACJA
POMYSŁ
OGÓLNIE

OVER THE NET
GENIAS

SPACE ACE II BORF'S REVENGE

Wielu posiadaczy Amigi zachwycało się programem SPACE ACE, który to przypominał raczej film animowany niż grę komputerową. Ci, którzy ją ukończyli z pewnością z zadowoleniem przyjmą wiadomość o jej kontynuacji. Borf wrócił i Ty drogi graczu znów wcielasz się w postać kosmicznego superbohatera, który musi powstrzymać żądnego zemsty kosmicznego terrorystę. Zamieniony poprzednio w dziecko Borf zostaje przywrócony do dawnej postaci przez nowych sprzymierzeńców Goonów. Przez nich zostaje również uprowadzona Twoja przepiękna narzeczona Kimberly. Tak zaczyna się nasza nowa przygoda. Gra, jak każdy tego typu program szokuje wspaniałą grafiką i cudownym dźwiękiem. Jest to animowany film, w którym gramy główną rolę. Jednak sama gra jest bardzo schematyczna i monotonna. O wiele więcej satysfakcji daje nam oglądanie pięknych obrazków i dobrej animacji niż sam w niej udział. Pokazując ten program można nieźle przedstawić możliwości Amigi, sama jednak akcja jest mało wciągająca. Program zajmuje sześć dysków i działa z rozszerzeniem pamięci (1MB).

DOSTATECZNY

GRAFIKA
DŹWIĘK
MOTYWACJA
POMYSŁ
OGÓLNIE

SPACE ACE II
BORF'S REVENGE
READYSOFT

Amiga i muzyka

AMIGA

BIBLIOTEKA AMIGOWCA

AMIGOWY Public Domain